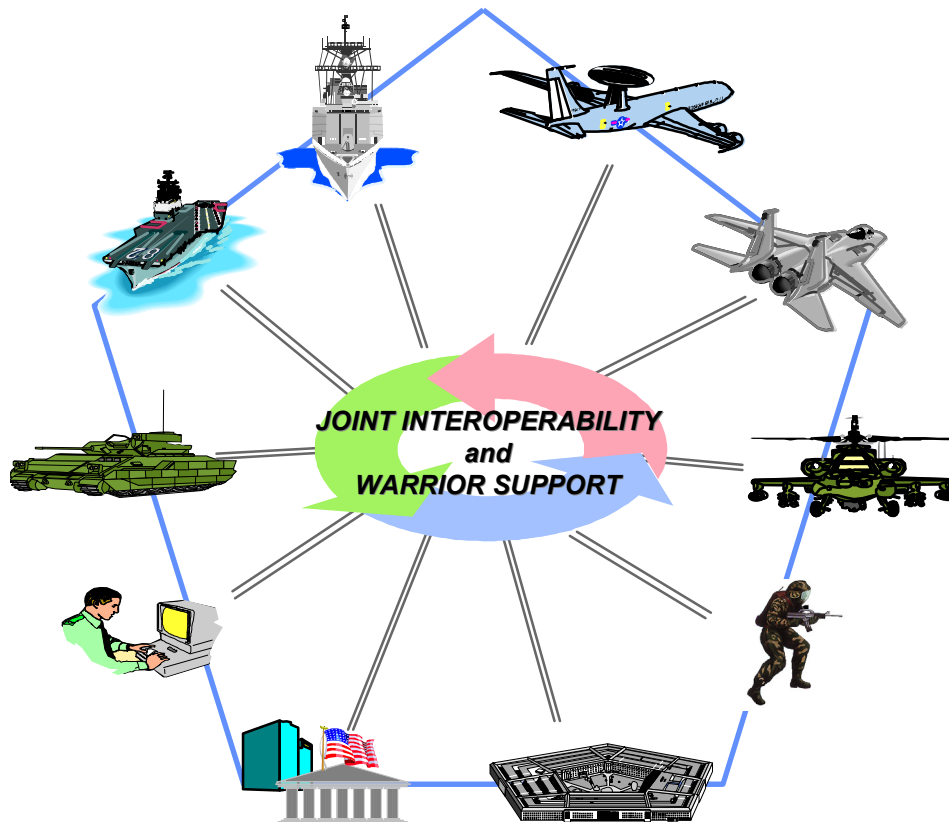


Department of Defense Technical Reference Model User Guide



Version 2.0

7 March 2003

Page Intentionally left blank.

Preface

Wherever possible, efforts were made to utilize the same style guide and form that drive the JTA document. Comment templates for updating the document were also derived from, and are basically the same as, those used in managing the JTA document. For more information on the TRM User Guide, see the DoD TRM web site: <http://www-trm.itsi.disa.mil>.

These addresses will provide additional information on documents cited in the DoD TRM and the DoD TRM User Guide. In addition, the DoD TRM web site has direct active links to these sites.

SAE	http://www.sae.org
C4ISR	http://www.c3i.osd.mil/org/cio/i3/AWG_Digital_Library.index.html
JTADG	http://www-jta.itsi.disa.mil/jta/jtadg/index.html (password required)
DII COE	http://diicoe.disa.mil
TASG	http://www-jta.itsi.disa.mil/jta/tasg/index.html (password required)
WSTAWG	http://wstawg.army.mil/

Page intentionally left blank.

Table of Contents

Preface	i
Section 1: Introduction to the User Guide.....	1
Section 2: Purpose	3
Section 3: When to Use the DoD TRM.....	7
Section 4: How to Use the DoD Technical Reference Model.....	9
Appendix A: Federal and DoD References	23
Appendix B: DoD TRM Services Summary	31
Appendix C: Frequently Asked Questions	45
Appendix D: Mapping	49
Appendix E: Case Studies	53
Case Study 1: Naval Fires Control System (NFCS)	55
Case Study 2: C4ISR Battle Group/Amphibious Ready Group (BG/ARG) Functional Design	66
Case Study 3: Transmission Control Protocol/Internet Protocol (TCP/IP) to TRM ..	70
Case Study 4: Mapping of Defense Information Infrastructure (DII) Common Operating Environment (COE) Segments to DoD Technical Reference Model (TRM) Services ...	78
Case Study 5: VETRONICS Technology Testbed (VTT)	81
Case Study 6: USPTO Technical Reference Model Case Study	91
Case Study 7: Mapping of JTA Version 3.1 Core Standards to TRM Version 1.0 Services and Interfaces.....	101
Case Study 8: Migrating Stovepipe Systems to Integrated/Interoperable Platforms Using the Technical Reference Model and Object-Oriented Operational Architectures	110
Case Study 9: LPD 17 San Antonio Class	129

Table of Contents

Section 1: Introduction to the User Guide

The DoD Technical Reference Model (DoD TRM) User Guide is to be used with the DoD TRM document. The User Guide provides added insight into a number of areas that are not elaborated in the DoD TRM document:

- How to use the DoD Technical Reference Model
- Insight into examples and case studies
- Different applications of the DoD TRM
- How to interpret and use model service and interface categories
- Contrasts and identifies the relationships between the DoD TRM document and other related documents

(e.g., Joint Technical Architecture [JTA], Defense Information Infrastructure Common Operating Environment [DII COE]; Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance Architecture Framework [C4ISR AF])

- Methodology for applying the DoD TRM

The examples and case studies provided in this guide are summary in nature and represent an encapsulated view of the interoperability and open systems issues addressed. In the interests of keeping the User Guide manageable and due to the extensive elaboration of certain service/interface descriptions contained in the DoD TRM, service and interface descriptions are also abbreviated. The DoD TRM should be referenced for the complete definition of a service or interface. DoD Memorandum April 15, 2002, Subject-Promulgation of DoD Technical Reference Model (TRM), Version 2.0 and DoD Memorandum August 12, 2002, Subject—DoD Joint Technical Architecture Version 4.0 contained in [Appendix A](#) should be consulted for amplifying guidance.

Page intentionally left blank.

Section 2: Purpose

The DoD TRM User Guide is promulgated to provide knowledge and insight in using the DoD TRM to address and resolve a variety of interoperability, portability, and open system issues. The purpose is to impart an understanding of how the model provides a foundation for developing technical and operational architectures, for defining services and interfaces, and when to invoke or use a particular model view (i.e., service or interface or both). Use of the DoD TRM and this guide promotes the development and fielding of systems that will support joint and combined operations interoperability, as well as information systems interoperability.

In the past, interoperability between DoD systems has not been addressed in a uniform and consistent manner in seeking effective solutions. The importance of battlefield interoperability and the ability of systems to exchange information is recognized as a decisive advantage in military operations and mandated in key DoD policy, regulations, memoranda, and directives: 5000.1 and .2-R; DoD 4630.5 and .8R (see [Appendix A](#)). Under revised policy mandates (CJCSI 3170.01B), interoperability has been defined as a key performance parameter (KPP), and must also be considered in mission unique systems since their products or components may be used elsewhere in the battlefield.

The model is not an end in itself nor is it an architecture. It is an aid to developing architectures and addressing a broad range of interoperability and open system issues. Additionally, the model can be used to support reuse and portability issues that are often intertwined with interoperability in the development of architectures, migration of systems, and legacy systems.

The scope of the DoD TRM is sufficiently broad to assist in addressing a wide range of problems and system configurations. The model does not restrict a user to specific system architectures, but rather supports distributed, networked, multi-tiered, single and multi-platform configurations and variants thereof.

A major theme to be reiterated throughout this guide and the DoD TRM document itself is that:

While the model is not formally mandated in all acquisitions, consistent use of service and interface definitions contained in the DoD TRM document is essential if interoperability is to be achieved. In this manner the warfighter can achieve a higher degree of interoperability across systems that can effectively fulfill mission requirements. Therefore, a key interoperability and open system requirement to be stressed in developments is to utilize the model's definitions first, before attempting to develop new ones. Use of model definitions and classes provides further assurance to other DoD stakeholders that a common foundation is being used across a broad range of DoD applications and the operational environment.

The DoD TRM is the foundation for both the JTA and DoD COE. Service categories contained in the latter two programs are derived from the DoD TRM definitions. Initiatives aimed at developing tailored or model variants thereof (e.g., functional model) should first draw from the structure and definitions contained in the DoD TRM in order to maintain the same consistency of service or interface definitions throughout DoD.

The more focused audience for use of the model falls into several basic categories:

A. Acquisition Users and Responders

Acquisition Users represent the developers of procurements, work packages and Requests for Proposals (RFPs), Statements of Work (SOW), source selection and contract support personnel. These individuals require knowledge of how to exploit the DoD TRM to aid them in making source selections and identifying effective deliverables in their solicitations. Contractors (Responders) responding to RFPs need to understand how to utilize the DoD TRM to make their proposals more responsive to interoperability requirements, and demonstrate how they understand the Government's needs.

B. Program Managers

Program Managers (PMs) who have interoperability and portability requirements levied on their system development or programs. PMs need to know how the DoD TRM can assist them in making informed decisions and trade-offs, how to satisfy requirements, and how keep their programs within cost and schedule. Current stovepipe funding methods create problems for Program Managers since additional funds to support interoperability requirements are not necessarily provided. PMs must negotiate with other PMs to resolve many interoperability issues. The more interfaces a system has in its operational environment, the more complex the interoperability issues.

It is recognized that PMs will not necessarily be the ones to directly involve themselves in many of these issues, but that their staff members will be designated as the action officers for interoperability issues. In many cases these technically oriented individuals are enlisted from the ranks of the laboratories that support the PMs and augment their staffs.

C. CIOs, Architects

Chief Information Officers and Architects include individuals that have requirements to: build different architecture views (i.e., technical and operational); develop architectural frameworks to support their organization's information technology infrastructure (e.g., standards profiles, reference models); select standards, technology and products. In order to structure their organization's infrastructure, knowledge of how to tailor and select appropriate model views, and knowledge of model views must all be known.

D. Technology/Technical Specialists

These individuals have requirements to address the selection of standards and technology, develop standards compliance plans, develop migration plans, make tradeoffs and resolve interoperability conflicts. These users must be able to utilize the DoD TRM to its fullest extent and work closely with CIO type personnel. These users include system engineers, technology officers, test personnel, and other technical specialists.

The above individuals can be found in both the Government and industry communities. These individuals must understand the relationships between the DoD TRM and other initiatives and models that rely on the DoD TRM as a foundation piece, such as the Joint Technical Architecture and Defense Information Infrastructure, and other programs such as the DoD (C4ISR) Architecture Framework, and the Levels of Information Systems Interoperability (LISI) Model. [Clarifying Note: The DoD TRM is used prior to development of an architecture and assists in the development of architectures or standards profiles (i.e., before the fact). The LISI model, as opposed to the DoD TRM, is used once a system has been developed to establish its level(s) of interoperability (i.e., after the fact). The models complement each other and do not compete in the same functional space.]

Purpose

Page intentionally left blank.

Section 3: When to Use the DoD TRM

3.1 Applying the DoD TRM

The model is to be used when addressing the following interoperability issues:

- When consistent and extensive service and interface terminology is required to address or describe an interoperability issue
- When functional analysis is to be performed and similar functions must be compared, matched, assessed or evaluated with other functions either within the same system or between disparate systems
- When mappings of services and interfaces are to be performed for the purpose of comparing functionality, products or standards
- When addressing migration issues that require knowledge of existing functionality, services and interfaces
- In developing standards profiles that must be categorized against a set of services and interfaces
- In developing different architecture views (e.g., technical, operational, system)
- In performing standards assessments to determine the degree of similarity, difference, non-applicability, completeness, orthogonality, or conflict within a standard or across standards
- In assessing products for incorporation into a system or for replacement of system components
- In assessing new technologies relative to the services or interfaces provided, and those that are impacted by the new or replacement technology
- When tailored model views (domain specific models) are required to support an enterprise or weapon system functional area
- When a framework is needed to support diverse platform configurations (e.g., client-server, networked, single and multi-processor configurations) and a representation of the services provided and interfaces contained within them must be developed

The model can be used to address a range of technical architecture developments, interoperability, and open system issues. Further insight into architectural configurations can be found in the DoD Architecture Framework (DoD AF) document. In developing a technical architecture, the basic approach to using the model is to initially utilize the definitions and relationships established in the DoD TRM document rather than “invent” or develop new ones. When new services or interfaces are required, these should be forwarded to the TRMWG for consideration and incorporation into the next version of the DoD TRM document.

Rather than invent new service categories, users are urged to work within existing definitions (e.g., elaborate on lower-level or sub-services). For example, the many different types of design automation tools and features (sub-services) that can be included are too numerous to identify under “Engineering Support Services.” Similarly, the many different types of “Communications

Services” that can be included in this respective category are also too numerous to define, unless such elaboration is required in the architecture under development. However, new services and interfaces not defined within the DoD TRM may be identified and included in a tailored view.

A company specializing in object-oriented products and technology, for example, may elect to highlight all of the object services into one object service category for product development purposes. The object services described in the DoD TRM’s “Distributed Computing Services, User Interface Services, and Data Interchange Services,” can all be grouped into one major category called Object Services to suit their needs. This is legitimate as long as the following are observed:

1. Rationale for the establishment of such a class is clearly documented, in the tailored model description.
2. The definitions contained in the new class are those basically derived from or defined in the DoD TRM set of definitions.
3. The new service category is defined clearly and with the same service or interface conventions used in the DoD TRM (i.e., remain within the same entity level). Rationale for entity level deviations must also be documented.

A listing of DoD TRM services and interfaces is contained in [Appendix B](#). Answers to common questions about the TRM are in [Appendix C](#).

3.2 Establishing Common Links

Definition: “Interoperability is the ability of two or more systems or components to exchange and use information. The ability of systems, units, or forces to provide or receive services from other systems, units, or forces, and to use the services so interchanged to enable them to operate effectively together.”¹

If the optimum degree of interoperability or openness is to be achieved across the enterprise, an accurate and consistent means or conversion to common units across systems and functions must be found to enable effective comparisons. Translations to a common denominator would thus enable accurate comparisons between requirements and systems, enterprise and system functionality, system and interface standards, and any reusable products. These common denominators should be identified as early as possible prior to system implementation, (i.e., at the enterprise or system requirements level). The DoD TRM and its set of service and interface definitions provides a framework for establishing the common denominators required to enable comparisons between functions, standards, and hardware and software components.

Establishment of the common denominators, (i.e., DoD TRM common services and interfaces to be used in addressing an issue), would facilitate the identification of interoperability points in the enterprise, environment, and within the systems. An additional benefit of using the DoD TRM is the ability to decompose interfaces and services so as to converge on critical interoperability issues in a timely and consistent manner.

1. IEEE 610.12

Section 4: How to Use the DoD Technical Reference Model

The DoD TRM is to be used in conjunction with an architecture development methodology, and is methodology independent. The model can be used to address interoperability issues in an existing or legacy environment, as well as in a new system development. The activities defined within the three steps are sufficiently generic that they support the development and tailoring of operational, technical, and system level architectures into more specific organizational and functional architecture views. The approach presented in the steps also provides insight into the development of migration strategies, legacy system development, or system enhancement approaches.

Within DoD and industry a number of standards-based architecture methodologies¹ can be found that address interoperability and open system issues. While the number of methodology steps may vary across them, the overall set of work tasks and activities contained within them address more or less the same issues. The DoD Architecture Framework defines a coordinated approach for DoD architecture development, presentation and integration², and as such, these architecture products have been added to the TRM three step methodology as reference to aid in using the DoD TRM and DoD Architecture Framework simultaneously. The steps in these methodologies also elaborate on architecture implementation, evolution, and maintenance. The 3 Step TRM Architecture Methodology defined in this section ([Figure 4-1](#)) is focused on the use of the DoD TRM only and is subordinate to the more extensive architecture methodologies such as those identified in [footnote \(1\)](#).

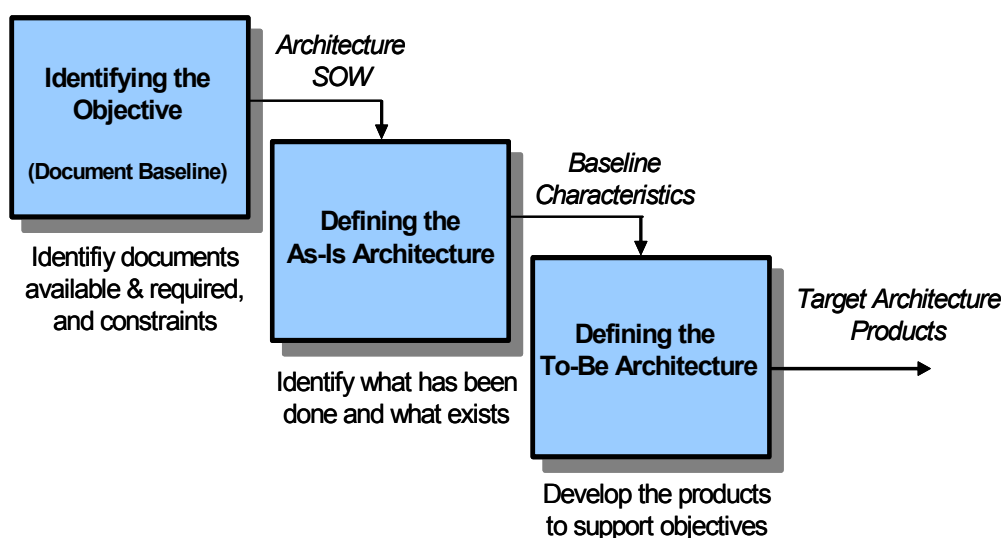


Figure 4-1. Three (3) Step Methodology for Applying the DoD Model

1. Examples of architecture methodologies: The Open Group Architecture Foundation (TOGAF), Version 5.0, December 1999; DoD's C4ISR Architecture Framework, Version 2.0, December 1997.
2. DoD Architecture Framework Version 2.1; October 2000.

The 3 Step TRM Methodology consists of:

- A [Step 1](#) to identify specific documentation needed to support the specific architecture objective, for example, baselining an “as-is” architecture, developing a migration plan or technical architecture
- A [Step 2](#) to represent the actual task of developing an “as-is” architecture and baselining what exists, while providing the input to the final Step
- A [Step 3](#) to develop and define the target architecture.

The architecture process is part of the overall development or acquisition process to ensure that interoperability is supported. Where possible the existing products or processes defined in the DoD 5000 series acquisition documents should be used to capture much of the architectural elements. The comparative and trade-off analyses of [Step 3](#), for example, can be combined with the systems engineering analyses that accompany any development activity. Additionally, DoD Architecture Framework products from existing systems should be used where they are available and applicable to creating a new architecture. In cases where such documentation is unavailable on existing systems, these architecture products may be required, and should be created during the appropriate step in the methodology to ensure the most effective use of the DoD TRM.

4.1 Step 1—Applying the Model ([Figure 4-2](#))

[Step 1](#) is a basic information gathering activity and must be performed in all cases of architecture development (i.e., existing or new system developments). Architectural methodologies used in conjunction with a reference model also require the same type of information in their initial steps (e.g., TOGAF, C4ISR AF or DoD AF). [Step 1](#) is independent of any milestone decision.

The purpose of this step is to identify and document the specific interoperability, open system, portability or standards task objective(s) to be accomplished. For example, is the objective to develop a technical architecture, develop a migration strategy or compliance plan, to address a standards conflict issue or develop a standards profile?

To properly support the objectives and ensure that the system(s) impacted is properly developed, the objectives must be implemented within a known operational environment and a sound acquisition or development process. The objectives are thus tempered by the known set of DoD requirements drivers that provide system justification (e.g., ORD, JROC). The latter requirements and operational environment establish the interoperability boundaries within which the system must function. Knowledge of the other respective DoD systems that must be interfaced can also be derived from operational requirements documents.

Accompanying the DoD requirements set are the acquisition policy and directives driving the system development (e.g., 5000 series, 4630 series, CJCSI 3170). The latter policy, mandates, directives and instructions provide insight as to the most appropriate points to effect some of the objectives. For example, Comparative and Trade-off Analysis of [Step 3](#) can be combined with or leverage system engineering analysis dictated by the particular milestone phase or program development activity.

Methodology for Applying the DoD TRM

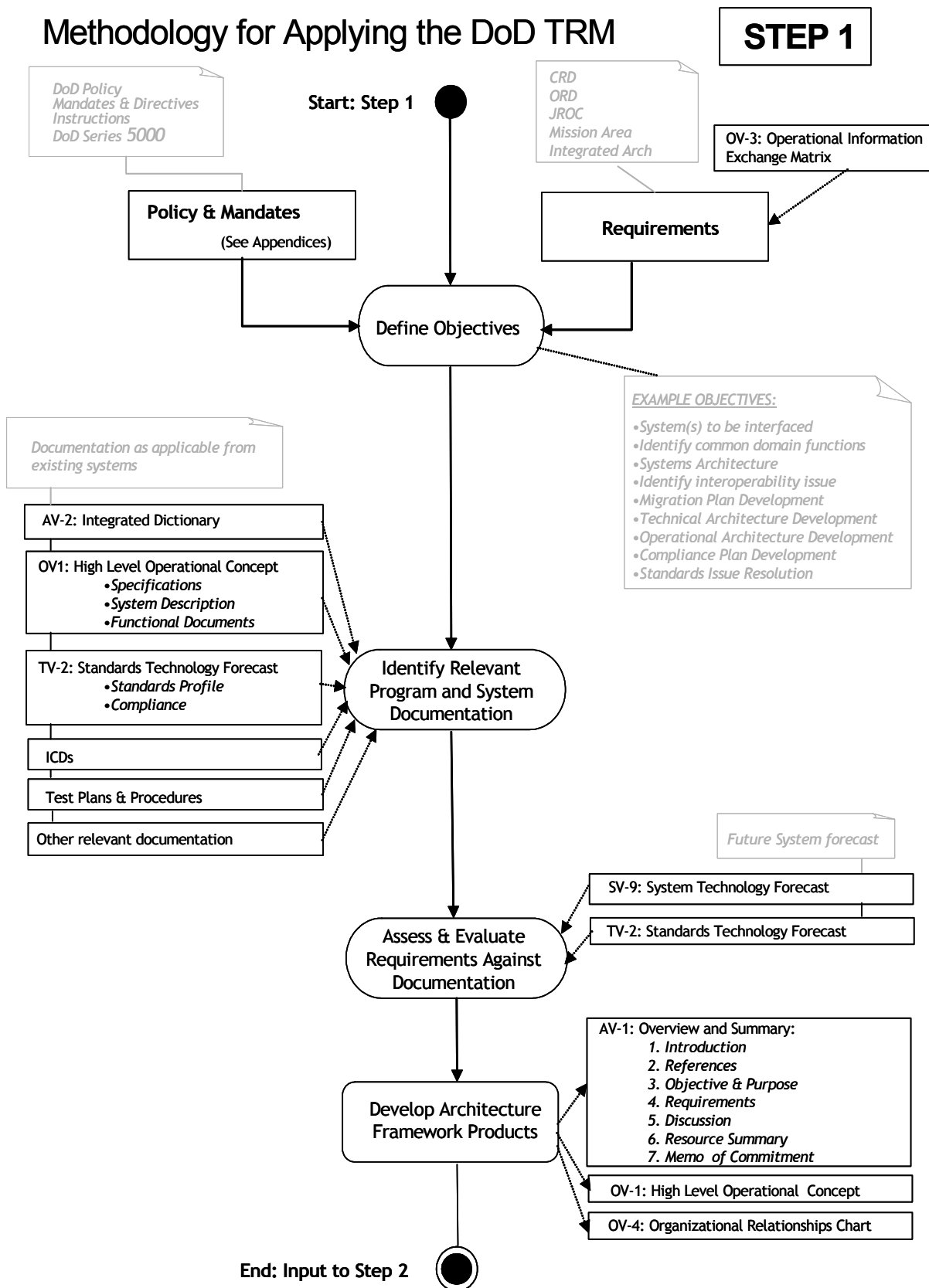


Figure 4-2.

Successful implementation of objectives requires knowledge of the system itself. Thus, relevant design specifications, interface control documents, functional specifications and known standards profiles for example, must be identified as part of the relevant system information. Such information can be obtained or derived through documentation of the existing system. For example, DoD Architecture Framework products which may support this discovery include:

Overview and Summary Information (AV-1) (used here from an existing system) defines the scope, purpose, intended users, environment depicted, and analytical findings and serves two purposes. In the initial phases of architecture development it serves as a planning guide, upon completion it provides summary textual information concerning “who, what, when, why, and how.”

The High Level Operational Concept Description (OV-1) (used here from an existing system,) provides a high-level graphical and textual description of operational concept and is the most general of all the DoD AF products. Its main purpose is to facilitate communication to high-level decision makers, and give an idea of the players and actions involved, and include graphics as well as explanatory text.

The Organizational Relationships (OV-4) document (from existing system ownership,) illustrates the command, control, coordination and other relationships among organizations or resources in an architecture which indicate fundamental roles and influence connectivity.

The Standards Technology Forecast (TV-2) (used here from an existing system,) provides a description of emerging standards that are expected to apply to the given architecture, and are anticipated to be available in a given set of timeframes (short (0–6 months), mid (6–18 months) and long (18+ months)).¹

The System Technology Forecast (SV-9) (used here from an existing system,) communicates emerging technologies (software/hardware) and industry trends, that are anticipated to be available in a given set of timeframes (short (0–6 months), mid (6–18 months) and long (18+ months)), that will affect future development of the architecture. It also includes confidence factors for the predictions.

The Operational Information Exchange Matrix (OV-3) (from an existing system,) provides the information that is exchanged between nodes and the relevant attributes of that exchange such as media, quality, quantity and the level of interoperability required.

These and similar documents provide knowledge of what was implemented, or is to be implemented, and thus represent additional constraints that must be taken into consideration.

With all of this knowledge in hand, objectives can be properly assessed and put in context relative to the interoperability requirements to be satisfied. Most of the documentation described exists to a large extent. In an existing system greater documentation should be available to facilitate decision making, but this is not the case necessarily. The proper assessment and evaluation of requirements and objectives thus enables the Acquisition or Program Manager to make a more informed decision. With interoperability identified as a Key Performance Parameter² greater

1. Summaries of DoD AF products are taken from DoD Architecture Framework, Version 2.1; October 2000

2. CJCSI 3170.01A Publication

accountability of a CIO and subsequently program managers is now established. The results of the assessment and evaluation can be properly documented in an Architecture Statement of Work. For example, in an existing system that is to be made JTA compliant, a detailed migration plan with a management commitment (Memorandum of Understanding [MOU]) supporting the degree of compliance can serve as the equivalent of an Architecture SOW. In the latter example, the justification for making system enhancements that require a standards waiver, would be included as part of the A-SOW or migration plan justification. Similarly, rationale for selecting an emerging standard would be included in the A-SOW.

The Architecture SOW must also reflect the degree of commitment by management that the objectives are to be satisfied in terms of effort, time and resources. The cost of implementing interoperability can be costly, impacting and time consuming on an acquisition or development. The degree to which interoperability will be supported must be clearly stated up front and represents a key output of [Step 1](#). Without this commitment by management successful implementation of subsequent steps is jeopardized. The Architecture SOW is different than the system or program SOW that is driving the contractor. It would be most beneficial if the A-SOW can be merged into the Program's SOW to provide the information as early as possible to the contractor or implementor. In many cases interoperability is implemented incrementally. In these cases the A-SOW would identify the increments and what is to be implemented per increment. In a joint program the interoperability agreements, responsibilities, schedules, and commitments of each Program Manager are identified to ensure successful implementation between the specific programs or systems. How these interoperability agreements fit into an overall or larger operational scenario would also be identified as part of the A-SOW. The DoD Architecture Framework AV-1 product is an effective example of an architecture SOW:

The DoD AF Overview and Summary Information (AV-1) defines the scope, purpose, intended users, environment depicted, and analytical findings and serves two purposes. In the initial phases of architecture development it serves as a planning guide, upon completion it provides summary textual information concerning "who, what, when, why, and how."¹

Below are recommended components of an architecture SOW:

Architecture SOW Outline

1. Objective and Purpose of the work being undertaken.
2. Policy References that identify or reinforce the requirements.
3. Requirements identification of the foundations for the work to be accomplished, a justification for the work to be done.
4. A references section. This section contains an identification of the program or project documentation that is available to provide insight into the system (e.g., specifications, technical parameters and constraint descriptions, interface and functional descriptions, operational scenarios). Documentation enables the identification of standards, services, system limitations, and existing technologies implemented in the system. In

1. Summaries of DoD AF products are taken from DoD Architecture Framework, Version 2.1; October 2000

the absence of such documentation, technical interchanges must be established between system-knowledgeable personnel and architecture developers to ensure that a consensus is reached on both the “as-is” and “to-be” architectures. This step can be a time-consuming step in both the negotiations (to arrive at a consensus) and in finding system subject matter experts.

5. Compliance Plan Outline or specific deliverable itself to be developed or identified as part of the A-SOW.
6. Summary of resources available to support work and development of the “to-be” system. This topic can be addressed via a Memorandum of Understanding (MOU).

4.2 Step 2—Developing the As-Is Architecture ([Figure 4-3](#))

[Step 1](#) represents an objectives identification and characterization, while the [Step 2](#) view is focused on identifying the “as-is” architecture (where applicable) and performing the essential mappings of services and interfaces to known functions and capabilities. [Step 2](#) is performed on a specific system that may or may not exist. In a new system, mappings are done directly to the DoD TRM. In the case of an existing system, additional effort is required to identify the previous model used. The initial [Step 2](#) activity is to identify any enterprise issues that represent constraints on the system or specific objectives identified in [Step 1](#) that require further refinement due to the environment. The enterprise issues may identify constraints that are imposed due to a broader need (e.g., imposition of a software development environment, language or technology). The state of a current system may be relayed through their DoD AF products, specifically the Systems View products. These products provide insight as to existing systems and their evolution over time.

The following DoD AF products may be developed in [Step 2](#), in additions to the inputs noted from [Step 1](#):

System Interface Description (SV-1) document identifies the systems and its components, their interfaces both within and between nodes. Documentation can be illustrated in three perspectives internodal, intranodal, and intrasystem.

Systems Communications Description (SV-2) illustrates the specific communications systems pathways or networks and details their configurations through which the physical nodes and systems interface. This is represented by an internodal and/or intranodal graphical perspective.

Systems² Matrix (SV-3) shows the system-to-system relationships in a given architecture. The matrix format is used to illustrate the internodal and intranodal relationships in rows and columns by recording the intersections based on characteristic such as status, category and classification level.

Systems Functionality Description (SV-4) focuses on describing the functions and the functional data flows among the system functions. It is the systems view counterpart to the Activity Model (OV-5) of the Operational View.

Methodology for Applying the DoD TRM

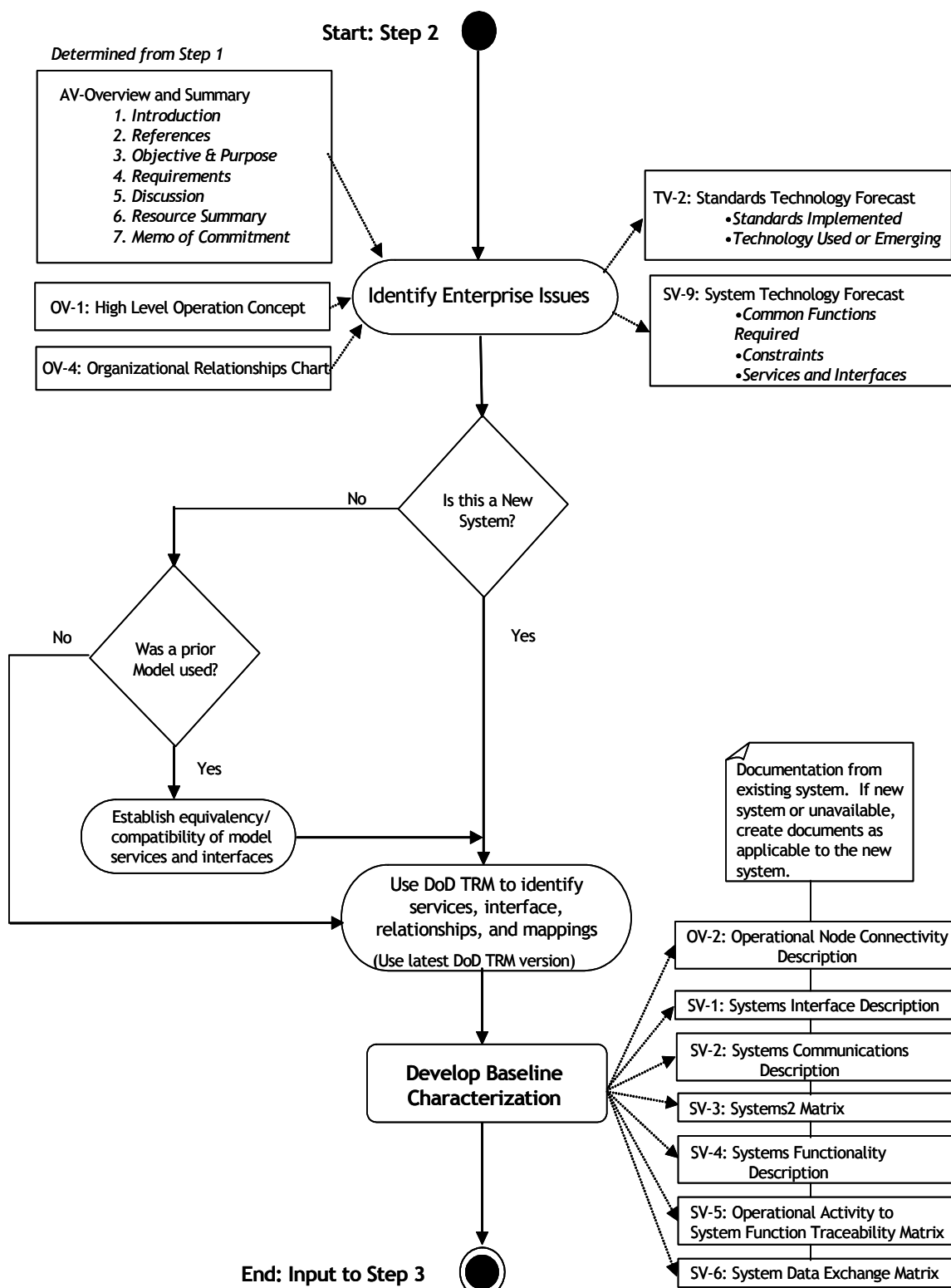
STEP 2

Figure 4-3.

Operational Activity to System Function Traceability Matrix (SV-5) details the mapping of the system functions back to the operational activities in a “many-to-many” relationship. Thus identifies the transformation of an operational need in to a purposeful action performed by a system component.

System Data Exchange Matrix (SV-6) details data exchanges among system elements, applications and H/W allocated to system elements. The focus of SV-6 is how the data exchanges ARE or WILL BE implemented, in system specific detail (i.e., protocols, data/media formats.)

System Performance Parameters Matrix (SV-7) describes the current performance characteristics of the system, and the performance requirements for all system(s) (hardware and software) at specified times in the future.

System Evolution Description (SV-8) describes the planned incremental steps toward migrating a suite of systems to a more efficient suite, or toward evolving the current system to a future implementation over time. SV-8 typically involves and evolution (greater scope, increased functionality/flexibility) of the system or migration (incrementally streamlining systems for efficiency and lower cost).

System Technology Forecast (SV-9) communicates emerging technologies (software/hardware) and industry trends, that are anticipated to be available in a given set of timeframes (short (0–6 months), mid (6–18 months) and long (18+ months)), that will affect future development of the architecture. It also includes confidence factors for the predictions.¹

Operational Node Connectivity Description (OV-2) describes the operational (both physical and abstract/virtual) nodes, activities performed at each node, connectivity and information flows between nodes with needlines to convey information transfers

Other enterprise constraints may require the identification of common functions that must be horizontally integrated across systems to facilitate the adoption of a common software element (e.g., DoD COE Product) or other reusable product. Once the enterprise issues are identified and assessed, specific “as-is” system architecture issues are focused on.

Subsequent to the identification of enterprise issues, the question is asked, “Is this a new system to be developed or not”. If it is not a new system a determination of the reference model used, if any, has to be made. The specific model used provides insight into the definition of services and interfaces, and their relationships. This is required to establish compatibility with the DoD TRM. Once done, insight is gained as to how to transition to the DoD TRM so as to converge on the use of a single reference model. If a model was not previously used, then DoD TRM definitions can be immediately identified that support the existing system (to the extent it can be done). Some tailoring may be required in definitions to accommodate the existing services and interfaces. The rationale or approach used to performing this assessment is to be included in the resulting Baseline Characterization document of [Step 2](#). New systems can directly use the DoD TRM to identify services and interfaces.

1. Summaries of DoD AF products are taken from DoD Architecture Framework, Version 2.1; October 2000

Once system services and interfaces are identified, the DoD TRM is used to develop the appropriate set of mapping matrices that correspond, for example, to a standard, product or other system to be interfaced. Mappings provide common denominators that enable equivalent comparisons between products or information. Services are identified in [Appendix B](#), and mappings are further described in [Appendix D](#), and represent a means of organizing information to enable the appropriate comparisons of such things as standards, common or reusable products, common functions across systems, technology to standards, and products to standards. Additionally, the DoD AF SV-3 product can be useful in this mapping exercise. The Systems² Matrix (SV-2 product) shows system-to-system relationships in a given architecture. The matrix format is used to illustrate the internodal and intranodal relationships in rows and columns by recording the intersections based on characteristic such as status, category and classification level¹. This information is captured in the baseline characterization document that serves as the input to [Step 3](#).

In essence, mappings enable a standard or product, for example, to be classified or grouped into a service or interface definition category. The association of a service or interface with a standard thus enables a comparison of other similar standards grouped into the same service to establish equivalency of the standards (i.e., same, orthogonal, or overlap), to resolve a standards conflict issue, or to address voids in the standard.

[Step 2](#) serves to capture the “as-is” architecture, when it exists, and documents it in the baseline characterization document. [Step 3](#) serves to define the “to-be” architecture and to identify what is required to achieve the target architecture. When an existing baseline characterization is not necessary (i.e., in the case of a new system) then much of the system mapping that would occur in [Step 3](#) is done in [Step 2](#). The result of this action is to leave [Step 3](#) primarily for performing additional tailoring or trade-off analysis that may be required to complete the target architecture activities.

4.3 Step 3—Developing or Completing the To-Be Architecture ([Figure 4-4](#))

Before a target or “to-be” architecture can be defined in its entirety several analyses must be performed. In the case of an existing system, additional or expanded services, interfaces or functions, beyond those identified in the baseline characterization document, must be identified. Once the complete set of functions, services, and interfaces are identified a determination must be made of the ability of the existing model definitions to accommodate them. Further tailoring may be required in cases where the system’s function definition is at a higher level of abstraction, and must be decomposed to a lower level so as to coincide (be map-able) into the model definitions. Development of the target architecture may require development of a tailored view. For example, a system command-control query function, may decompose into a message transmission service, a database access and management service, and require a messaging and journaling support service.

1. Summaries of DoD AF products are taken from DoD Architecture Framework, Version 2.1; April 2000

Methodology for Applying the DoD TRM

STEP 3

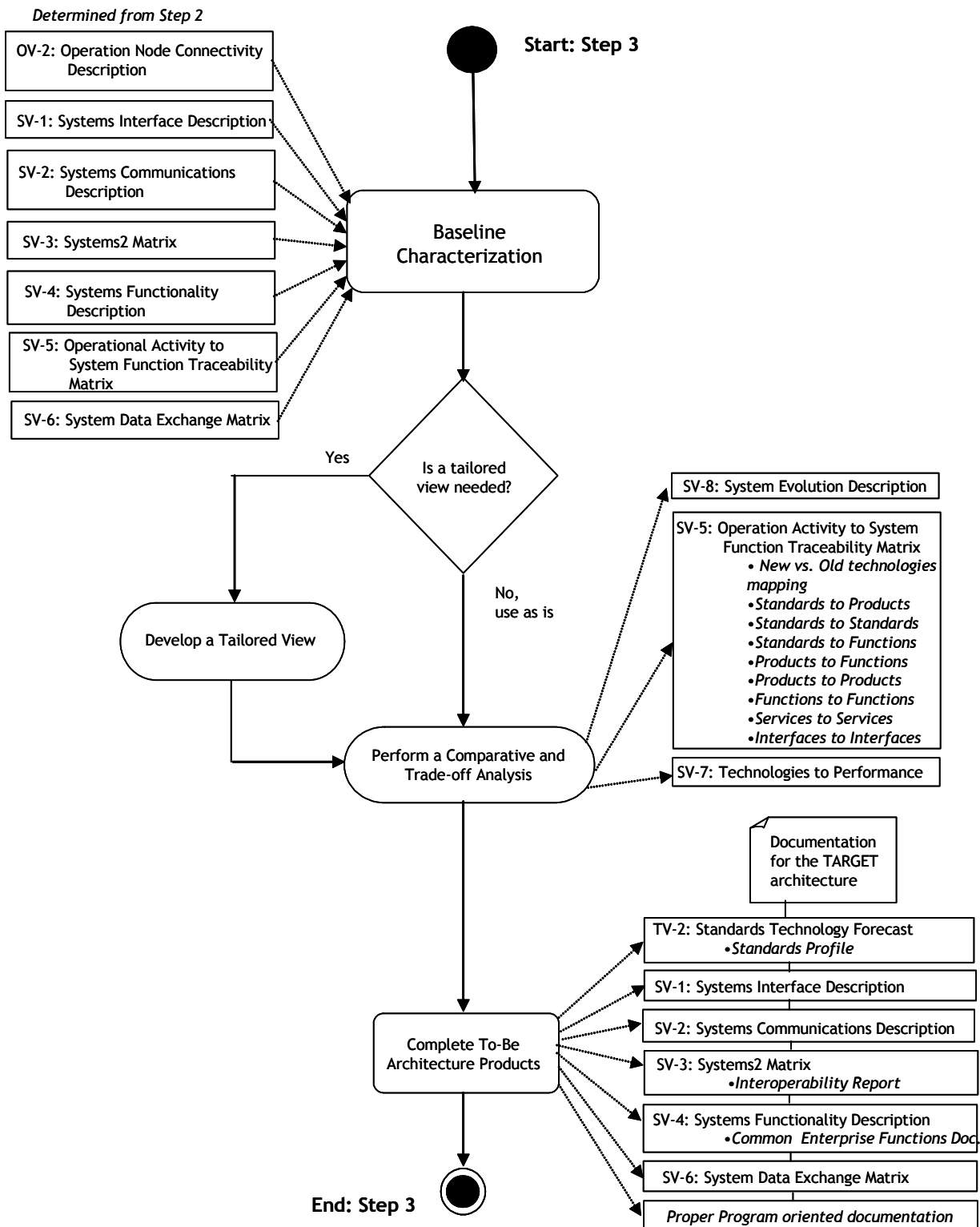


Figure 4-4.

The DoD Architecture framework provides several products to assist in developing the “to-be” architecture. Specifically, the Systems View products provide a representation of systems and their interrelationships providing for or sustaining warfighting missions. There is only one mandatory DoD AF systems view product, the (SV-1) Systems Interface Description. Its purpose is to describe in various levels, the details that may be necessary within three perspectives (internodal, intranodal, and intrasystem) of discrete architectures.

The DoD AF products described in [Step 1](#) and [Step 2](#) will also be developed in [Step 3](#), however, they will be developed for the “to-be” system, and assist in supporting the documentation of the overall program.

A tailored view may result in the development of a model variant (referred to in some cases as a functional reference model) to highlight a set of functions or capabilities that have been defined within the users domain of familiar or working definitions. Care must be exercised in tailoring a model or model service definition. A distinction should also be made between the need for a new or separate service category, and including a set of services as subordinate to an existing service or interface definition. In the event a clear distinction of where a service belongs cannot be made using the DoD TRM, consultation with the TRMWG for resolution or clarification is recommended. There are additional DoD Architecture Framework product which may aide in determining if a tailored view is required. For example, the Systems² (i.e., system-to-system relationship) Matrix (SV-3) and the Systems Functionality Description (SV-4) support identification of interoperability issues, as wells as descriptions of common enterprise functions respectively.

Once [Step 2](#) and [Step 3](#) tailoring are completed, a set of mappings exist that relate products, standards, or system services and interfaces to model services and interfaces. At this time attention can be focused on performing comparative and trade-off analysis. With analysis and the mapping of standards, products or other items into model definitions and categories, the following can be performed (see [Appendix D](#)):

- New and old technologies can be compared as to their impact and appropriate phase-in. Associated standards can also be identified and incorporated into any resulting standards profiles.
- Standards can be compared to determine if they overlap, conflict, or cover the same subject matter or from a different perspective (orthogonal). In some cases, the particular subject area may not be covered at all pointing to a deficiency.
- Products can be assessed to determine whether they conform to a particular standard or not. Relevant DoD AF products may be assessed at this stage.
- Product to product comparisons are facilitated when mapped into their corresponding services and interfaces.

The results of [Step 3](#), e.g., “to-be” architecture, information for developing a standards profile, interoperability definition, are then documented taking into consideration the results of the analysis and how they impact the specific architecture issue at hand.

The completion of service and interface identification for the “to-be,” architecture together with the “as-is” architecture information of [Step 2](#), enables the performance of cost-effectiveness trade-off analysis. These trade-off analyses are an integral part of the interoperability report produced as part of [Step 3](#). In some architecture methodologies this analysis is known as a “gap analysis”.

Summary of the Three Steps

[Step 1](#):

- Identify major objectives
- Identify relevant DoD policy, directives, instructions, etc.
- Identify key DoD requirements drivers from MNS, CRDs, ORDs, JROC, etc.
- Identify relevant program documentation, e.g., DoD AF Products, specifications, ICDs, system descriptions
- Assess and evaluate objectives against requirements
- Develop Architecture Statement-of Work and Memorandum of Understanding that identifies degree of commitment to interoperability issue

[Step 2](#):

- Identify enterprise issues (new or existing system)
- Identify reference model used (existing system)
- Identify existing services and interfaces (existing system)
- Identify services and interfaces (new system)
- Perform mappings (new or existing system)
- Develop Baseline Characterization (new or existing system) following appropriate framework, e.g., DoD Architecture framework

[Step 3](#):

- Identify complete set of functions, services and interfaces (new or additional ones)
- Develop tailored view (if needed)
- Perform comparative and trade-off analyses
- Document Target Architecture or architecture issue within appropriate architecture framework

Beyond [Step 2](#) and [Step 3](#)

Subsequent steps in various methodologies focus on the prioritization of the added or new functionality to be implemented as a function of cost, schedule, and other benefits. Subsequent methodology steps include migration planning that allows the sequencing of work tasks or incremental development. Actual architecture development subsequently begins and is followed by maintenance tasks that provide for the continuous monitoring of such items as changes in technology, doctrine, or environment changes. The DoD Architecture framework provides additional architecture products to support further development of new architectures and supporting the evolution of existing systems.

Any methodology employed in developing architectures must be sufficiently flexible and generic to support a broad range of applications if interoperability is to be properly served. In the DoD environment, weapon systems coexist alongside communications, business, logistics, and other support systems that in many cases must process the same information between them or a transformed derivative of the same, all part of the larger DoD information infrastructure. Examples of how the TRM is applied to actual systems development are in [Appendix D](#).

Page intentionally left blank.

Appendix A: Federal and DoD References

The following Federal and DoD references are provided as information sources that reinforce the mandate for interoperability and considerations that should be given to such. The interoperability and open system issues are pervasive throughout the Federal Government. A top down order is presented whereby Federal mandates are identified first, followed by a linking or threading down to DoD specific mandates, implementation instructions and policy. The list is representative and not all inclusive. The various individuals involved in developing architectures or addressing interoperability should consult these references for rationale or supporting documentation in justifying their need for use of a technical reference model. The listed information includes extracts and briefs of larger documents that should be consulted for additional and amplifying information.

OMB Memorandum A-130 for the Heads of Executive Departments and Agencies

Guidance on the development and implementation of Information Technology Architectures. Required in the Clinger Cohen Act of 1996.

“An architecture in compliance with the Clinger Cohen Act and OMB guidance will contain two elements:

1. The Enterprise Architecture
2. A Technical Reference Model and Standards Profiles.”

Other extracts:

“The TRM and Standards Profiles comprise a cross-cutting element, affecting all components of the Enterprise Architecture. Standards enable interoperability, portability, and scalability in systems throughout the agency.”

“The TRM identifies and describes the information services used throughout the agency. The standards profile defines a set of IT standards that support the services articulated in the TRM; they are the cornerstone of interoperability. Standards Profiles are the published sets of standards or the source references for standards that prescribe the interfaces between those services that will be standards-based.”

Memorandum for Defense Acquisition Community August 1996

Subject: Update of the DoD 5000 Documents

5000.1 articulates general principles to guide all defense acquisition programs.

5000.2-R establishes mandatory procedures for Mandatory Procedures for Major Defense Acquisitions Programs (MDAPs) and Major Automated Information Systems (MAIS) acquisition programs (and selected other programs).

DoD 5000.1 Summary

Applies to all elements of DoD.

Highly sensitive classified programs, cryptologic, and intelligence programs shall follow guidance in this Directive.

MAIS Acquisition Program:

AIS Acquisition program that:

- Is designated by ASD (C3I) as a MAIS
- Has estimated program costs in excess of \$30 million in fiscal year 1996 constant dollars; total program costs in excess of \$120 million in FY 1996 constant dollars; or total life cycle costs in excess of \$360 million in FY 1996 constant dollars.

MAIS—additionally, for the purpose of determining whether an AIS is an AMIS, the following shall be aggregated and considered a single AIS:

- The separate AISs that constitute a multi-element program
- The separate AISs that make up an evolutionary or incrementally developed program or
- The separate AISs that make up a multi-component AIS system.

DoD 5000.2-R

Applies to MDAPs and MAIS Acquisition Programs and specifically where stated, less-than-major programs. In general, highly sensitive classified programs and cryptologic and intelligence programs shall follow the guidance of other programs.

For MDAP and MAIS definitions—see DoD 5000.1.

The policies and procedures described in DoDD 5000.1 and this Regulation are mandatory.

Open Systems: PMs shall address the use of open standards in the design of all systems elements (mechanical, electrical, software, etc.) The design effort shall select open standards for interfaces based on the criteria described in the open systems strategy. Interfaces are internal, external, physical and functional. Selected interfaces shall be controlled by standards adopted by recognized standards organizations whenever possible. When these standards are not effective, de facto standards (set by the market place) shall be used. This approach shall be followed to develop a standards-based architecture in designing systems. PMs shall document means for assuring conformance to open standards and determining the extent of openness of system, subsystems, and/or components at the levels specified (in paragraph 3.3.1 of 5000.1).

Interoperability: Compatibility, interoperability and integration are key goals that must be satisfactorily addressed for all acquisition programs. These goals shall be specified and validated during the requirements generation process. Satisfaction of these requirements shall be addressed throughout the acquisition life-cycle for all acquisition programs. The DoD JTA is mandatory for all emerging systems and systems upgrades. The JTA applies to all Command, Control,

Communications, Computers and Intelligence (C4I) and automated information systems, and the interfaces of other key assets (e.g., weapons systems, sensors) with C4I systems. The Component Acquisition Executive may grant waivers to the standards in the JTA with the concurrence of the USD (A&T) and the ASD (C3I). Interoperability of C4I Systems shall be in compliance with DoDD 4630.5, DoDI 4630.8, DoDI 4630.8, CJCSI 6212.01A and CJCSI 3170.01A. (CCA and PRA)

DoD 4630.5—Information Interoperability

Applies to OSD, Military Departments, Chairman JCS, Combatant Commands, and the Defense Agencies.

Applies to any information system that produces, uses, or exchanges information in any form electronically; that crosses a functional, Component, or combined boundary; and that produces an operational capability for the warfighter or other DoD decision-maker including:

- New systems, legacy systems, modifications to existing systems and communications and computer interfaces to and among weapon systems;
- DoD communications and information technology interfaces to the National Foreign Intelligence Programs, the Tactical Intelligence and Related Activities, and the Joint Military Intelligence programs;
- Systems developed or acquired through any acquisition method including Advanced Concept Technology Demonstrations (ACTD), and other acquisitions under section 2371 of the United States Code.

Systems described above shall be compliant with the JTA.

DoD information systems shall be interoperable with other DoD systems.

Draft 4630.8-R—Information Interoperability

This regulation assigns responsibilities, and prescribes mandatory procedures to achieve Interoperability of the Defense Information Infrastructure (DII) and tactical Signals Intelligence (SIGINT) systems. All emerging systems and systems upgrades shall comply with the DoD JTA. Existing systems are to migrate to the applicable JTA standards, while considering cost, schedule, and performance impacts.

A System Standards Profile and a summary list of all systems that must interoperate the system shall be developed for each system during Program Definition and Risk Reduction Phase and provided to the Joint Staff for interoperability approval.

The DoD components shall ensure that the JTA is implemented, including enforcement, budgeting, and determining the pace of systems upgrades.

CJCSI 3170.01B

The DoD CIO is responsible to ensure the interoperability of information technology and national security systems throughout the DoD.

This revision reflects major reformat of the document; major changes include document submission for Automated Information systems, substantive update to the CRD enclosure and format, substantive update to the ORD enclosure and format, mandates Interoperability Key Performance Parameters for CRDs and ORDs and denies time-phased requirements in support of evolutionary acquisition, addresses program affordability for ORDs, defines US Atlantic Command role for interoperability, and clarification of definitions.

KPP—Key Performance Parameter:

- A. That capability or characteristic so significant that failure to meet the threshold can be cause for the concept or system selection to be reevaluated or the program to be reassessed or terminated. KPPs are extracted from the ORD and included in the Acquisition Program Baseline. User or user-representative participation in each acquisition phase is essential. (DoD Regulation 5000.2-R)
- B. Those performance parameters validated by the Joint Requirements Oversight Council (JROC) and included in the Acquisition Program Baseline (APB). They are capabilities or characteristics considered most essential for successful mission accomplishment. Failure to meet a KPP threshold can be cause for the concept or system selection to be reevaluated or the program to be reassessed or terminated. (CJCSI 3170.01)

CJCSI 6212.01B

Applies to the Military Services, Joint Staff, combatant commands, and those activities reporting to the Chairman of the Joint Chiefs of Staff.

Applies to the compatibility, interoperability, and integration of new or modifications to existing DOD systems that have C4I capabilities (including weapons systems, DOD Foreign Intelligence Programs and Tactical Intelligence and Related Activities) acquired or developed in support of military operations.

Applies to automated information systems (AISs) not normally included in C4I definitions but which have missions requiring interface to the joint warfighter. Highly sensitive, classified programs will comply with this CJCSI but will be tailored as necessary to account for special security considerations.

The overall objective of this CJCSI is to develop, acquire and deploy C4I systems and equipment that meet essential operational needs of US forces and are interoperable with existing C4I systems and equipment.

The DOD components shall implement the JTA to ensure that interoperability requirements are reflected in the requirements documents.

DoD Technical Reference Model, Version 2.0

The DoD Technical Reference Model (DoD TRM) Version 2.0 Promulgation Letter and document represents DoD's response to the need for a technical reference model. The Letter recognizes the need for a TRM and satisfies the mandate of A-130.



CHIEF INFORMATION OFFICER

DEPARTMENT OF DEFENSE
6000 DEFENSE PENTAGON
WASHINGTON, DC 20301-6000

April 15, 2002

MEMORANDUM FOR SECRETARIES OF THE MILITARY DEPARTMENTS
CHAIRMAN OF THE JOINT CHIEFS OF STAFF
UNDER SECRETARIES OF DEFENSE
DIRECTOR, DEFENSE RESEARCH AND
ENGINEERING
ASSISTANT SECRETARIES OF DEFENSE
GENERAL COUNSEL OF THE DEPARTMENT OF
DEFENSE
DIRECTOR, OPERATIONAL TEST AND
EVALUATION
DIRECTORS OF THE DEFENSE AGENCIES
DIRECTORS OF THE DOD FIELD ACTIVITIES
DIRECTOR, JOINT STAFF

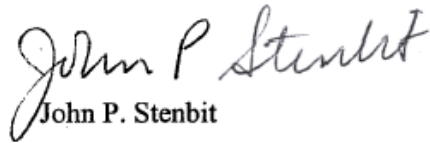
SUBJECT: Promulgation of DoD Technical Reference Model (TRM) Version 2.0.

This memorandum is effective immediately and transmits guidance to DoD Services, Agencies, and Combatant Commands on the use of the DoD Technical Reference Model Version 2.0. This memorandum makes the DoD TRM Version 2.0 effective for immediate use superseding Version 1.0. The DoD TRM defines and describes Global Information Grid (GIG) services and interfaces, and their relationships to be used in supporting the development of integrated architectures, the DoD Joint Technical Architecture, C4ISR Architecture Framework, and Common Operating Environment.

Each DoD Component and cognizant OSD authority is responsible for deriving their respective technical reference model view, where required, based on the DoD TRM structure. This action will ensure consistent use of service and interface definitions where interoperability issues must be addressed and effectively managed as well as promote software portability and reuse.

Addressees should assure the widest distribution of this memorandum. Request Director Joint Staff forward this memorandum to the Unified Combatant Commands.

The DoD TRM and related information can be found at URL <http://www-trm.itsi.disa.mil>.



John P. Stenbit

Office of the Secretary of Defense
1000 Defense Pentagon
Washington, DC 20301-1000

Aug. 12, 2002

MEMORANDUM FOR SECRETARIES OF THE MILITARY DEPARTMENTS
CHAIRMAN OF THE JOINT CHIEFS OF STAFF
UNDER SECRETARIES OF DEFENSE
GENERAL COUNSEL OF THE DEPARTMENT OF
DEFENSE
INSPECTOR GENERAL OF THE DEPARTMENT OF
DEFENSE
DIRECTOR, OPERATIONAL TEST AND EVALUATION
ASSISTANTS TO THE SECRETARY OF DEFENSE
DIRECTORS OF THE DEFENSE AGENCIES
DIRECTOR, JOINT STAFF
DIRECTORS OF THE DOD FIELD ACTIVITIES

SUBJECT: DoD Joint Technical Architecture Version 4.0

The Joint Technical Architecture (JTA) continues to evolve as technology and standards change and remains one of the important tools for achieving improved interoperability and other DoD goals. This memorandum approves JTA Version 4.0 for use effective immediately, superseding JTA Version 3.1. The DoD Components are reminded that the responsibilities and procedures for the development and use of the JTA are contained in DoD 5000.2-R dated April 5, 2002, and DoD I 4630.8 dated May 2, 2002.

Addressees are requested to assure the widest distribution of this memorandum. Request Director, Joint Staff forward this memorandum to the Combatant Commands. The JTA and related information can be found at [URL http://www-jta.itsi.disa.mil](http://www-jta.itsi.disa.mil).

/s/

E.C. Aldridge, Jr.
Under Secretary of Defense
(Acquisition, Technology &
Logistics)

/s/

John P. Stenbit
Department of Defense
Chief Information Officer

Appendix B: DoD TRM Services Summary

Within each major entity services are listed alphabetically.

B.1 Services Summary

Applications Software

User Software Applications:

- User specific
- System specific
- Application specific

Support Applications:

Business Processing:

- Calendar
- Calculation
- Project Management
- Spreadsheet

Communications Applications:

- Broadcast
- Communications Conferencing
- Enhanced Telephony
- Organizational Messaging
- Personal Messaging
- Shared Screen Teleconferencing
- Video Teleconferencing

Database Utilities:

- Networking/Concurrent Access Services
- Query Processing
- Report Generation
- Screen Generation

Engineering Support:

- Computer-aided Design
- Decision Support
- Expert System
- Modeling and Simulation

Environment Management:

- Batch Processing
- Computer-based Training
- Information Presentation and Distribution
- Transaction Processing

Multimedia:

- Audio Processing
- Document Processing
- Electronic Publishing
- Image Processing
- Map Graphics
- Multimedia Processing
- Video Processing
- Text Processing

Application Platform Services

System Services:

Data Interchange Services:

- Characters and Symbols
- Compression
- DoD Applications
- Document Interchange
- Hardware Applications
- Mapping
- Optical Digital Technologies
- Product Data Interchange
- Raster/Image Data Interchange
- Technical Data Interchange

Data Management Services:

- Data Dictionary/Directory
- Database Management System
- Transaction Processing

Distributed Computing Services:

- Client-Server
- Object
- Remote Access

Graphics Services:

- Device Interfaces
- Raster Graphics
- Vector Graphics

Internationalization Services:

- Character Sets and Data Representation
- Cultural Convention
- Native Language Support

Platform Communications Services:

- Networks
- Application-oriented
- Subnetwork Technologies
- Transport-oriented

Security Services:

- Access Control
- Architectures and Applications
- Authentication
- Availability
- Confidentiality
- Integrity
- Non-repudiation
- Security Labeling
- System Management

Software Engineering Services:

- Bindings
- Computer-aided Software Engineering tools & environment
- Language
- Software Life Cycle Processes

System Management Services:

- Configuration Control
- Fault Monitoring
- Information System Security Management
- Other Management
- Performance Monitoring
- State Management
- Usage Monitoring and Cost Allocation
- User/Group Management

User Interface Services:

- Character-based User Interface
- Graphical Client-Server
- Object Definition and Management
- User Interface
- Window Management

Operating System Services:

- Clock/Calendar
- Fault Management
- Kernel Operations
- Media Handling
- Operating System Object
- Real-time Extension
- Shell and Utilities

Physical Environment Services:

- Devices
- Hardware Processing

External Environment Services

- Devices
- Communications Infrastructure (e.g., telephones, networks, cabling, packet switching equipment)
- Systems
- User (physical/cognitive)

B.2 Services Detailed (Expanded Definitions)

B.2.1 Support Applications Services

Business Processing:	Common office functions used in day-to-day operations
Calendar	Manage personal tasks and time, coordinate schedules
Calculation	Routine and complex arithmetic calculations
Project Management	Tools that support planning, administration, and management of projects
Spreadsheet	Capability to create, manipulate, present information in various forms
Communications Applications:	Capability to send, receive, forward, manage electronic and voice messages
Broadcast	One-way audio, audio/video communications services
Computer Conferencing	Conference via workstations, document exchanges
Enhanced Telephony	Call forward, call waiting, programmed directories, teleconferencing, voice mail
Organizational Messaging	Send, receive, forward, display, retrieve, validate, disseminate, prioritize, manage, authenticate
Personal Messaging	Send, receive, forward, store, display, manage personal messages, et al.
Shared-Screen Teleconference	For two or more users using shared workstation windows
Video Teleconferencing	Two way video, full motion display

Database Utilities:	Capabilities to retrieve, organize, and manipulate data from DBMSs
Networking/Concurrent Access Services	Manage concurrent user access to DBMS
Query Processing	Interactive selection, extraction, and formatting of information
Report Generation	To define and generate hard-copy reports from DBs
Screen Generation	To define and generate screens that support retrieval and presentation, update of data
Engineering Support:	For analyses, design, modeling, development, simulation for users and environments. Includes CAD services, decision support tools, expert-system shells
Computer-Aided Design	High-precision drawing and modeling tools
Decision Support	Interactive modeling and simulation tools for analyzing alternative decisions
Expert System	Artificial intelligence capabilities based on knowledge or inference engines
Modeling and Simulation	Capabilities to capture or set object characteristics
Environment Management:	Integrate and manage the execution of platform services
Batch Processing	Capability to queue work, sequence managing, asynchronous tasks
Computer-Based Training	Provide integrated training environment, online documentation, help files, context sensitive definitions
Information Presentation and Distribution	To manage distribution and presentation of information, can store, archive, prioritize, restrict, recreate information
Transaction Processing	On line capture and processing of information in an interactive exchange with the user

Multimedia:	Manipulate and manage information consisting of text, graphics, images, video, audio
Audio Processing	Capture, compose, and edit audio information
Document Processing	Create, edit, merge, format documents, scanning
Electronic Publishing	Photographic images, color graphics, advanced formatting and style features
Image Processing	Capture, scan, create, and edit images
Map Graphics	Map manipulating, creating entity symbology, create, edit, compose drawings, symbols, maps
Multimedia Processing	Compress, store, retrieve, modify, sort, search, print hypermedia, optical storage technology, data compression, digital storage techniques
Video Processing	Capture, compose, edit still graphics, title generation,
Text Processing	Create, edit, merge, and format text

B.2.2 Application Platform Services

Data Interchange Services:	Support for interchange of information between applications and external environment
Characters and Symbols	Interchange of character sets/fonts/date/time representation
Compression	Algorithms for data (text, still images and motion mages) storage and exchange
DoD Applications	Functional areas unique to DoD mission that are not standardized
Document Interchange	Specifications for encoding data (text, pictures, numerics, special characters), and logical/physical structure of documents
Hardware Applications	For data interchange between non-homogeneous hardware, bar-coding, optical disk-handling, graphics device interface
Mapping	Formats and facilities for machine-readable graphics-based mapping, charting and geospatial data
Optical Digital Technologies	For light and optical technologies to capture, encode/decode, store data
Product Data Interchange	Specifications that describe drawings, documentation and data for product design and manufacturing, for geometric and non-geometric data
Raster/Image Data Interchange	Handling/manipulating raster graphics and images, pixel-by-pixel representation or imagery data exchange and attachments to images
Technical Data Interchange	Standards for interchange of graphics data, vector graphics, technical specifications

Data Management Services:	Management of data independent of processes that create or use it
Data Dictionary/Directory	Define and obtain data in DB, access and modify data, internal/external formats, integrity, security rules, standardization and registration of data elements, data sharing and interoperability
Database Management System	Data management in a distributed system, data administration, controlled access to, and modification of, structured data, create, populate, move, backup, restore, and archive DBs
Transaction Processing	Support for online capture and processing of information in interactive exchange with the user
Distributed-Computing Services:	Support for applications distributed or dispersed among systems in a network yet maintain cooperative processing environment
Client-Server	Computing services partitioned into requesting processes (clients) and providing processes (servers) on same/distributed platforms
Object	Definition, instantiation, interaction of objects in a distributed environment, OS bindings, message transport and delivery, data persistence
Remote Access	Location transparency functionality, access to appropriate systems resources (files, data, processes)
Graphics Services:	Standards for creating and manipulating pictures
Device Interfaces	Services for accessing graphics devices (monitors/printers)
Raster Graphics	Image representation on a matrix of dots, monochrome/gray-scale/color bit maps, creation by scanner, cameras, and color paint software packages
Vector Graphics	Graphical objects as sets of end points for lines, curves, and geometric shapes, geometric knowledge and display lists, shape integrity

Internationalization Services:	Services and interfaces to allow users to define, select, change between culturally related application environments, different market segments
Character Sets and Data Representation	Input, store, manipulate, retrieve, communicate, data independent of coding scheme, GUI modifications, syntax-consistent-semantic-independent
Cultural Convention	Support for local rules and conventions
Native Language Support	Support for more than one language for local character sets
Platform Communications Services:	To support distributed applications requiring data access and applications interoperability
Networks	<i>(NEED DEFINITION HERE)</i>
Application-Oriented	Functions and interfaces on network and communications system protocol software used by applications
Subnetwork Technologies	LANS and other data communications services concerned with physical and data-link layers (1and2 OSI Model)
Transport-Oriented	End-to-end transmission of data across network and end to end reliability, end-end error detection/recovery, flow control and monitoring quality of service

Security Services:	For protection and separation of sensitive information
Access Control	Unauthorized use of information-system resources
Architectures and Applications	Security architecture and placement of security into specific applications
Authentication	Unique and proper identification and authentication of system elements
Availability	Assurance of timely and regular communications, graceful degradation
Confidentiality	Ensures that data is not made available to unauthorized individuals or computer processes
Integrity	Protection of system through open-system integrity, network integrity and data integrity, data is not altered or destroyed in an unauthorized manner
Non-Repudiation	Non-denial of origin or delivery of data, validation of source software packages and hardware
Security Labeling	Accuracy and integrity of security labeling
System Management	Certification, accreditation, and risk management, alarm reporting, audits, cryptographic key management
Software Engineering Services:	For development and maintenance tools of software applications
Bindings	Access to applications-bindings and object code linking
Computer-Aided Software Engineering Environment (CASE) Tools	Requirements specifications and analysis, design, testing, and prototyping, configuration control
Language	Syntax, semantic definitions, shell/script/procedural/object-oriented/3 rd -generation languages
Software Life-Cycle Processes	Activities, methods, processes, transformations, development/maintenance, all phases including post-deployment support

System Management Services:	State management, configuration control, performance, monitoring, fault monitoring, user/group management, usage monitoring
Configuration Control	Identification, control, status accounting, verification, software distribution, license management
Fault Monitoring	Event management and network recovery, loss or incorrect operation of system components
Information System Security Management	Installation, maintenance, enforcement of information domain and system security policy rules
Other Management	Database management and administration, print management
Performance Monitoring	Performance aspects of hardware, software, network components, system resource management, device management
State Management	Monitoring, maintaining, and changing state of system
Usage Monitoring and Cost Allocation	Management of licensing, system cost management, system resource allocation
User/Group Management	Interfaces for administering users and groups, implementation of management policies across a system, group/user access to applications
User Interface Services:	How users interact with an application
Character-Based User Interface	Command line, menu-driven, keyboard input, no graphics
Graphical Client-Server	Relationships between client and server processes, graphical user interface display processes
Object Definition and Management	Characteristics of display elements-color, shape, size, movement, graphics context
User Interface	Interaction with applications, how to gain access to applications programs/OS/utilities, i.e., menus, screen designs, keyboard Commands, command language, the way user interacts with computer
Window Management	How windows are created, moved, stored, retrieved, removed, and related to each other

Operating System Services:	Core services to operate application platform and provide an interface between application software and platform
Clock/Calendar	Mechanisms for measuring and maintaining all time
Fault Management	Prevention, isolation, diagnosis, correction whenever abnormalities occur
Kernel Operations	Low-level services to create/manage processes, execute programs, manage files/directories, and control I/O processing to and from peripheral devices
Media Handling	Disk and tape formatting of data and interchange of data
Operating System Object	Rules for creating, deleting, managing objects
Real-Time Extension	Support for event-driven processes, interrupt processing
Shell and Utilities	Operator-level services-comparing, printing, displaying file contents, file management, sorting data, displaying file contents
Physical Environment Services:	
Devices	<i>(NEED DEFINITION HERE)</i>
Hardware Processing	<i>(NEED DEFINITION HERE)</i>

B.2.3 External Environment

External Environment:	Entities with which Application Platform exchanges information
Devices	Peripheral, External Hardware
Communications Infrastructure	Telephones, networks, cabling, packet-switching equipment
Systems	Simulations external to the System
User (physical/cognitive)	Physical, administrative, personnel; provide for security protection of information-system components in external environment, Human Machine Interface, Human Computer Interface

Page intentionally left blank.

Appendix C: Frequently Asked Questions

- Q1.** What is a technical architecture?
- A.** Recent discussions within DoD have defined three types of architectures: operational, technical, and system. A technical architecture is a set of rules, or “building codes,” that are used when a system engineer begins to design/specify a system. These rules consist primarily of a common set of standards/protocols to be used for sending and receiving information (information transfer standards such as Internet Protocol suite), for understanding the information (information content and format standards such as data elements, or image interpretation standards) and for processing that information. It also includes a common human-computer interface and “rules” for protecting the information (i.e., information system security standards).
- Q2.** Why is the DoD TRM needed?
- A.** The DoD TRM provides an extensive and comprehensive set of service and interface definitions that establish a common vocabulary for addressing open systems, portability, and interoperability issues. The model provides a basis for the identification, comparison, and selection of existing and emerging standards and their relationships.
- Q3.** What is the scope of the DoD TRM?
- A.** The scope of the DoD TRM includes information technology, weapon systems, and real-time systems in DoD systems that may exchange information services or share interfaces across a joint, functional, or organizational boundary.
- Q4.** What enhancements were made to the DoD TRM beyond the TAFIM TRM?
- A.** A defined set of logical and direct interfaces view was added to the DoD TRM. The interfaces view allows the model to support real-time interoperability issues and other weapon system interfaces.
- Q5.** What is the relationship of the DoD TRM to the SAE GOA Standard?
- A.** The interface definitions in the DoD TRM are consistent with the SAE GOA definitions. If an interface standard is to be cited, reference to the SAE GOA should pose no barriers with applications using the DoD TRM.
- Q6.** What impact will the DoD TRM have on prior TAFIM TRM applications?
- A.** None, the services view of the DoD TRM was adapted from the TAFIM TRM. Implementations using the TAFIM TRM can be migrated using the DoD TRM with virtually no impact for the same set of common services.
- Q7.** What type of systems does the DoD TRM support?
- A.** The DoD TRM can support a broad range of C4ISR and weapon systems. The range of applications, as contained in the definition of IT in the DoD TRM, can be supported by the DoD TRM, as well as other applications.

- Q8.** In a legacy system, if a previous particular reference model cannot be identified, what impact will using the DoD TRM have on new functionality, services and interfaces?
- A.** There should be no impact as long as the basic DoD TRM service or interface definitions are used independent of any platform configuration or model entity allocation.
- Q9.** What are the foundation models used in the JTA and DII COE?
- A.** The TAFIM TRM is the foundation model for these initiatives. Thus, there are no interoperability issues that present any barriers in using the DoD TRM.
- Q10.** Do I have to use both the service and interface views contained in the model?
- A.** No. The technical architecture or interoperability issue to be addressed will identify what view(s) is needed. Within a view, only those services or interfaces needed to support the interoperability issue are used. Additional services or interfaces can be added as needed.
- Q11.** What updates, if any, have been done to the JTA relative to the DoD TRM?
- A.** The DoD TRM has now been incorporated into the JTA, Version 3.0, replacing the TAFIM TRM. In addition, the DoD TRM is also used in the JTA Weapon System Annex.
- Q12.** Who has oversight of programs?
- A.** Program Executive Officers (PEOs) provide executive direction and oversight of assigned programs, and act as milestone decision authorities (MDA) for certain programs. PEOs will ensure that PMs have identified assessed, and implemented applicable architecture requirements.
- Q13.** Who is responsible for the execution of acquisition programs?
- A.** PMs have full authority, responsibility, and accountability for the execution of assigned acquisition program within their approved acquisition program baseline. PMs will include in all appropriate solicitations language mandating that products and services be compliant with the architecture. PMs will identify, plan and budget the necessary resources to support the architecture implementation efforts including compatibility, interoperability, and integration testing during the evaluation of systems and equipment.
- Q14.** What is the role of the Major Decision Authority (MDA)?
- A.** The MDA serves as the decision authority for assigned programs and ensures that programs have identified and implemented applicable technical architecture (TA) requirements. The MDA will be provided input from the CIO endorsing TA compliance.

- Q15.** Why should I do a Migration Plan and how long do I have to complete a Migration Plan?
- A.** The purpose of migration is to achieve the end-state of technical architecture compliance. The intent of the migration planning process is to achieve broad, early consensus on the approach to specific program migration. A goal of six months is established for completing the preparation, submission, review, and approval of a Migration Plan for a given program. See [Step 2](#) and [Step 3](#) (methodology) of Applying the DoD TRM in the User Guide for additional insight into migration plans.
- Q16.** Which programs need a Migration Plan?
- A.** All-current programs or capability the produces, uses, or exchanges information in any form electronically must develop, submit, and implement a Migration Plan.
- Q17.** Who approves the migration plan?
- A.** The review and approval of each specific system migration plan will be coordinated between the PEO and the CIO.
- Q18.** Does each program need an individual migration plan?
- A.** No. Where appropriate, multiple programs may be covered by a single migration plan. However the CIO must endorse each program independently for the specific program compliance(e.g., JTA, DII COE.
- Q19.** On what is the PEO approval based?
- A.** The PEO approval is based on maintaining program stability. The PEO will review and assess the program specific migration plan for the significance of architecture compliance (e.g., technical architecture) feasibility and impact, proposed actions, and overall program risk.
- Q20.** What kinds of architecture does the TRM support?
- A.** The TRM can support a variety of architectures (e.g., Technical, Operation and System). However, high level operational architecture functions will require decomposition into lower level function (see VTT case study).

Page intentionally left blank.

Appendix D: Mapping

Why perform a mapping?

Mappings enable comparisons between the following items:

1. Products
2. Standards
3. Systems
4. Products & standards
5. Systems & standards
6. Systems & products

for the purpose of addressing interoperability issues. These issues address the establishment of interface standards that apply in developments, identification of commonalities between products and standards, and identification of common functions and capabilities across user or operational domains.

Approach

Performing a relative mapping ([Figure D-1](#)): Item A is first mapped against model services or interfaces. Item B is then mapped against model services and interfaces. The resulting mappings are then overlaid to identify common intersection points in the mapping matrix. The overlaps represent common functions, services or interfaces that may be the same or similar across the entities (i.e., products, standards, etc.) being compared. Thus, using the DoD TRM services and interfaces as the common denominator comparisons or tradeoffs can be performed.

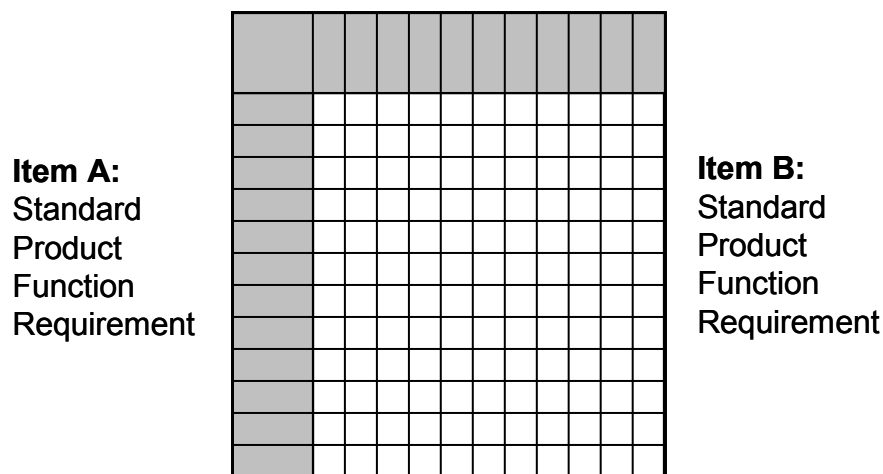


Figure D-1. Model Services & Interfaces

What does a mapping accomplish?

A mapping identifies common denominators across systems and other architectural views. Developing a series of mappings and identifying the intersecting elements provides common points to enable analyses, comparisons, and a more judicious selection of standards and products.

How is a mapping performed?

Mappings are performed using the entities, services and interface definitions of the DoD TRM. Mappings are developed by identifying system, product or standards services and interfaces and comparing them to the DoD TRM services and interfaces for equivalency. The specific type of mapping to be performed is defined in the [Step 1](#) Methodology of [Section 4](#) of this User Guide.

For example:

If two items are to be compared, the functions, services, or interfaces that each item contains or supports must be identified. For example, a system or database standard that contains information about the query language or command structure could map into the DoD TRM as shown in [Figure D-2](#). As can be seen, some of the System A services are sub- or lower-level services of others. According to the DoD TRM, Query Language and Data Dictionary are sub-services of Data Management, and TCP/IP is a sub-service of Communications Services.

System A Services: A(n)		DoD TRM Services: S_(M)									
		S ₁ Software Engineering*	S ₂ Data-Interchange	S ₃ Engineering Support*	S ₄ Communications	S ₅ Business Processing	S ₆ User Interface	S ₇ Data-Management	S ₈ Data Utilities	S ₉ System Management	S ₁₀ Graphics
A1-Business Processing					X						
A2-Mapping		X									
A3-Query Language							X				
A4-Data Dictionary							X				
A5-Raster & Vector										X	
A6-Report Generation							X				
A7-Protocols-TCP/IP			X								
A8-User Interface						X					
A9-System Management									X		

*Services not used in System A.

A1, A3, A4, A5, A6, and A7 are lower level services that are identified as contained within DoD TRM services S₂, S₄, S₇, and S₈ (see DoD TRM document definitions).

**Figure D-2. Mapping of System Services to DoD TRM Services
(Mapping Matrix A)**

Thus, a mapping of a system's or a standard's services $A(n)$ to DoD TRM services $S_{(m)}$, where $A1, A2, A3$, etc., are system services, and S_1, S_2, S_3 , etc., are model services, defines a common set or service relationships. Performing similar mappings of other systems or standards to the model services or interfaces would allow the following:

- A. Comparison of functions that are similar to or supported by both systems since they are both being mapped against the DoD TRM common set of definitions. This can be used to establish common functions across systems to address horizontal integration issues in an environment or across a DoD functional area.
- B. In a technical architecture document, e.g., JTA, standards associated with a particular service/interface can be identified and compared to those identified in an existing or proposed system for the same service/interface. The underlying assumption is that the JTA is mapped and developed against DoD TRM functions, and those of the system are similarly mapped. This is the concept of mapping against common denominators (i.e., the model service and interface definitions).
- C. In an exercise, a product via its supported or provided services can be mapped against the DoD TRM, and subsequently against a standard to examine compliance or against another product to address potential technology insertion or redundancy.

In [Figure D-3](#), Standard A supports services S1, S2, S3, S4, S5, S11, S12, S13, and S15 (see the DoD TRM for a complete listing of services) against Standard B supporting services S1, S2, S4, S5, S11, and S14. It can then be determined if one standard can replace the other or serve as a suitable standard until an emerging one is mature enough to be substituted. The additional or different set of services supported by a newer standard may result as a consequence of new technologies, extended support for new functions, services or interfaces. In any event, mappings reveal or make visible differences or similarities in those items being mapped and compared. In this example, Standard A may be a candidate for replacing Standard B.

		Standard B					
		S1	S2	S4	S5	S11	S14
Standard A	S1	X					
	S2		X				
	S3						
	S4			X			
	S5				X		
	S11					X	
	S12						
	S14						X
	S15						

**Figure D-3. Standard A to Standard B Comparison
(Standard A candidate to replace Standard B)**

This same mapping approach can be used when addressing interfaces. An example of an interface mapping ([Figure D-4](#)) against the model is presented to identify the types of interfaces and standards found within System A.

		DoD TRM Interfaces									
System A Standards		1L	1D	2L	2D	3L	3D	3X	4L	4D	4X
1D - EIA 232		X									
1D - EIA 422			X								
1D - Mil-STD 461											
2D - AM-7239(E)				X							
3L - ISO/IEC 9945					X						
3L - Mil-STD 188-220B						X					
4L - Mil-STD 2525a									X		

Figure D-4. Identification of Interface Types & Standards

Thus, via a series of mappings, different types of comparisons and analyses can be made to assist in addressing an interoperability issue or developing a technical architecture.

Sample Exercise:

Performing a mapping—The objective of this exercise is to identify, using the service or interface definitions of the DoD TRM, the set(s) of services or interfaces that are common to an item or set of items (e.g., standards, products, enterprise, system)

1. Service and interface definitions and relationships are derived from the DoD TRM.
2. Using system documentation (e.g., specifications, functional or system descriptions, interface control documents) identify the functions involved in the interoperability issue or problem to be addressed.
3. Once system or enterprise functions are identified, compare them to the DoD TRM services or interfaces to identify similar or equivalent definitions or relationships between them.
4. Develop a matrix of services or interfaces to identify similarities (mapping). One axis contains the system/enterprise services or interfaces, the other axis contains the DoD TRM ones.
5. Develop other matrices as required: If a product or standard is to be assessed then on one axis the product or standard services are included. Model services are identified on the other axis. An analysis of the product or standard must be done first to identify the set of services or interfaces associated with the item. The intersection of matrix elements represents common services or interfaces between the items being mapped.
6. Development of a set of matrices identifies the common areas that support the particular interoperability issue at hand.

Appendix E: Case Studies

A series of case studies using the DoD Technical Reference Model (TRM) are presented in encapsulated form. These case studies are intended to demonstrate the wide range of applications the DoD TRM can be applied to in addressing interoperability, portability, and open systems issues. The case studies illustrate several scenarios that are found in DoD systems and applications:

1. To address interoperability in a domain where both new systems and legacy systems/components are found. The Naval Fires Control System is a new system that must interoperate with other existing fire control systems of the Army and Marine Corps, and other forward observer devices.
2. The Battle Group/Amphibious Ready Group case study is one where key issues of concern are horizontal integration and identifying common functionality within a domain to address interoperability-in-the-large.
3. The TCP/IP case study is system independent and represents a communications issue addressed by the DoD TRM to address protocol issues.
4. The DII COE case study is focused at assisting the mapping of DII COE services and products against the model to identify potential portable/reusable products, or common products between systems. The complete study and mapping is quite voluminous. A similar mapping was accomplished between the DoD Joint Technical Architecture (JTA) and the DoD TRM to assist in the selection and comparison of standards used across different domains and systems (to be included in the future). The DoD TRM is currently being used to assist in the JTA restructuring.
5. The Army Tank Command's Vetronics Technology Testbed (VTT) Case Study utilizes both views of the DoD TRM (i.e., services and interfaces). This new weapon system concept illustrates the value and impact of developing an open system architecture, using the DoD TRM to define vehicle operator station interfaces, common functions, and services. The results of this Army research and case study can have an impact on how future vehicle operator stations and support functions are designed and implemented in the future. The case study presented is an encapsulation of a more extensive project at the US Army Tank-Automotive Research, Development and Engineering Center (TARDEC).
6. The Patent Trademark Office case study represents an example where the Technical Reference Model has been institutionalized by a Federal Agency to manage their IT assets. The case study also identifies cost savings and benefits that have already been accrued by the PTO over the past four years.
7. The Joint Technical Architecture (JTA) mapping to the DoD TRM represents another example (similar to the DII COE mapping). However, it is a mapping example where the JTA standards have been mapped to the model services and interfaces. Both the DII COE and JTA mappings illustrate how organizational or enterprise products and standards can be mapped to the DoD TRM, and subsequently managed and maintained.

8. The NORAD/USSPACOM case study demonstrates how the Technical Reference Model and an object-oriented operation view approach can be used to migrate disparate and stove-piped systems into an integrated and interoperable environment. The case study brings together the operational, system, and technical architecture views into a cohesive entity to address enterprise interoperability, reuse, and the command-control domain.
9. The Navy LPD 17 C4ISR Interoperability case study demonstrates how the Technical Reference Model can be used to assess systems interoperability, anticipate configuration and version compatibility issues, and aid technology transition and insertion, both within and across the LPD family of ships. With the ships' production/delivery schedule significantly longer than the technology insertion time, effective technology transition approaches become key to fielding consistent and accurate system configurations.

Readers should refer to the DoD TRM web site for additional case study information.

Case Studies and Examples

1. Naval Fires Control System
2. C4ISR Battle Group/Amphibious Ready Group Functional Design
3. TCP/IP to TRM Case Study
4. DII COE to TRM Mapping
5. VTT Case Study
6. USPTO Case Study
7. JTA to TRM Mapping
8. NORAD/USSPACECOM Case Study
9. Navy LPD 17 C4ISR Interoperability Case Study

Case Study 1: Naval Fires Control System (NFCS)

NAVSEA—Lloyd Lamont Designs, Inc.

Objective of the Case Study

The objective of this case study is to examine how the Technical Reference Model (TRM) may be used in conjunction with the Naval Fires Control System (NFCS) to improve the interoperability of the technical, system, and operational architectures of the system. NFCS consists of the interfaces, computer resources, software, human-computer interface, and personnel required to conduct fire support operations. NFCS performs the following functions in support of Naval surface fires missions:

- Provide situational awareness by accessing and presenting the tactical picture of the area of operations
- Prepare and disseminate fire plans from a target list
- Support the coordination and declinations of fire support missions with other force missions
- Command and monitor the execution of fire missions by ownship's weapons
- Maintain and report the readiness of naval surface fires systems to perform their mission

The TRM provides a foundation for defining a conceptual framework and a vocabulary to better coordinate the acquisition, development, and interoperability of DoD systems. This is accomplished through the definition of defined direct and logical interfaces, as well as the application of an extensive set of service definitions and relationships. This case study provides a general description of how NFCS relates to the TRM, and how the employment of the TRM can assist in the development and description of the NFCS technical architecture model. This study then illustrates the examination of a single service area in the formation of NFCS development plans.

Background

NFCS is a NAVSEA sponsored “new start” system which is being developed to provide mission planning and execution support for much of the Navy’s future Land Attack mission support. NFCS will be the key mission support system for providing target information to a number of new Land Attack weapons systems including:

- Extended Range Guided Munition (ERGM)—a long range (up to 60 nautical miles) rocket-assisted, GPS-guided 5-inch artillery round
- Land Attack Standard Missile (LASM)—a missile designed to extend Land Attack operations out to 100 nautical miles
- Tactical Tomahawk—a modification to the Tomahawk weapons system which introduces the concept of on-station loitering as the weapon awaits assignment to target

Rationale/Purpose

This case study illustrates a technical architecture for NFCS which was developed using the DoD TRM. The TRM provides a common conceptual framework, and defines a common vocabulary so that diverse components within DoD can better coordinate acquisition, development, interoperability, and support of DoD information systems. The TRM technical architecture view defines services within layers.

The TRM incorporates a set of interfaces that are based on the SAE General Open Architecture model, to define the interfaces between layers. These layers share information with each other through a defined set of protocols. The TRM stack (stacked layers) begins with the physical hardware layer and proceeds through a set of layers to the application layer via a precise set of direct and logical interface definitions that clearly identify the separation of boundaries between the layers.

A layered structure provides the following attributes: layer portability, interoperability between associated layers, plug-and-play implementations, and affordability via contractor-competitive competition for the development of each of the independent layers. The TRM also defines a set of services that can be used to describe the NFCS functions associated with communications, data transfers and routing, and human computer interfaces.

Description/Process Used

The process used in this study to develop the NFCS technical architecture consisted of examining written material and briefs describing NFCS, attending engineering team meetings, and analyzing the NFCS architecture data as compared to the guidance in the TRM. The key documents which were used to establish the systems current architectural status include the NFCS Operational Requirements Document (NFCS ORD), a draft of the NFCS System Specification, a draft NFCS performance specification, and numerous briefs and related Land Attack documentation.

NFCS is being developed to support Navy and Marine Corps operational concepts which require effective sea-based firepower to support sustained operations ashore and other operations in the littoral regions. The delivery of this firepower requires the development of an automated system for controlling naval fires that is:

1. Compatible with the Global Command and Control System—Maritime (GCCS-M)
2. Compliant with the Defense Information Infrastructure Common Operating Environment (DII COE)
3. Functionally interoperable with Marine Corps Command and Control (C2) systems

In the development of the NFCS technical architecture, a series of architectural development steps were undertaken. [Figure E-1](#) illustrates the first step which is the identification of the top level functions that must be supported to execute naval fire missions.

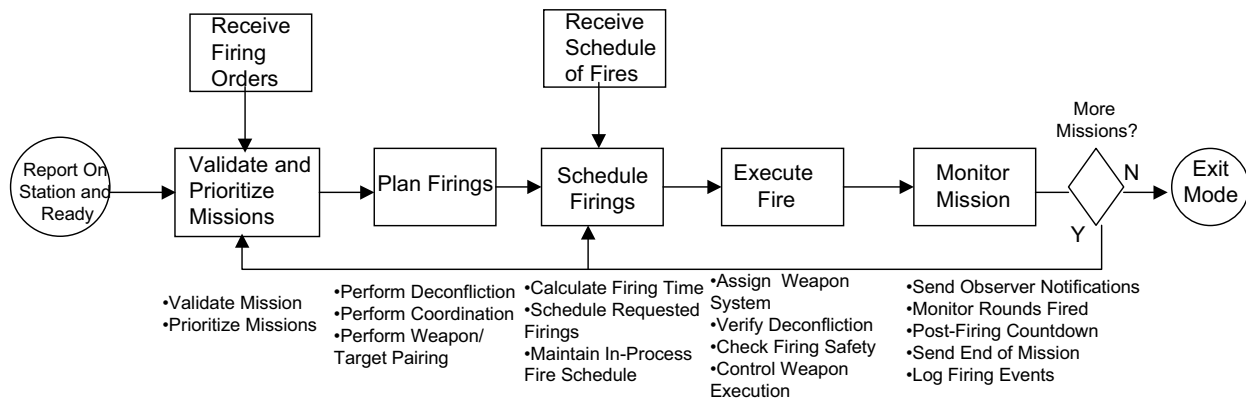


Figure E-1. Top Level Naval Fires Process

From the functions illustrated in [Figure E-1](#) a set of specific functions supported by NFCS has been developed. The following diagram ([Figure E-2](#)) illustrates the major functions of the NFCS system.

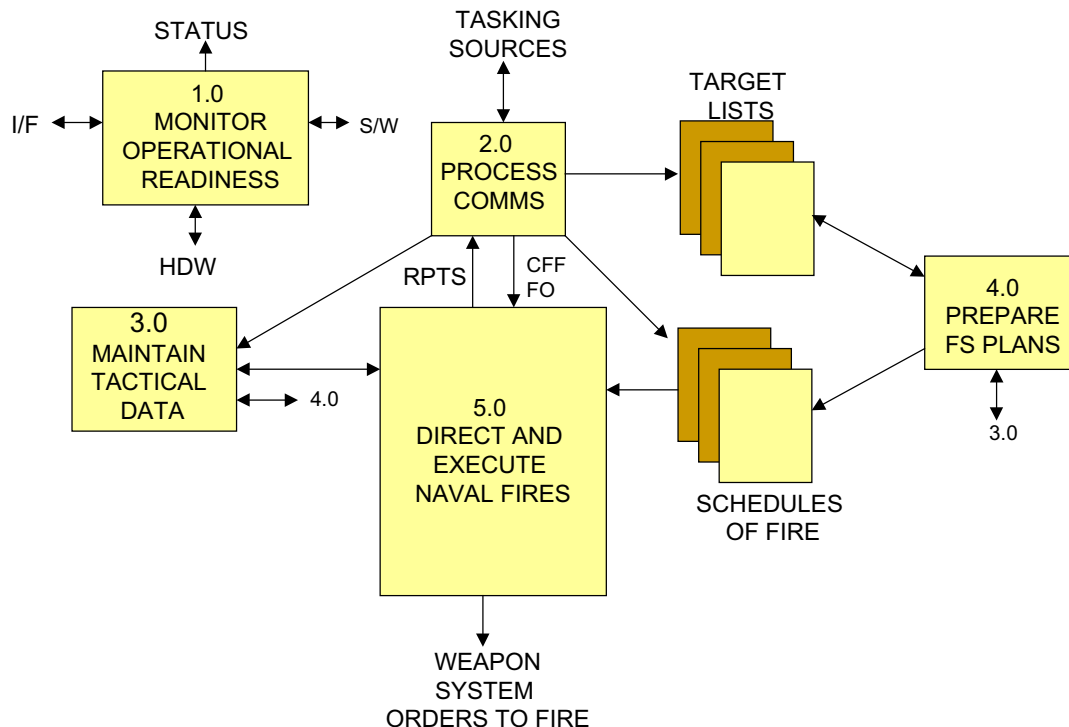


Figure E-2. Major NFCS Functions

The identification of the major functions performed by NFCS coupled with a basic understanding of the interfacing operational systems has lead to a notional architecture which identifies the major components of NFCS. In the diagram below ([Figure E-3](#)) the basic notional NFCS architecture is illustrated.

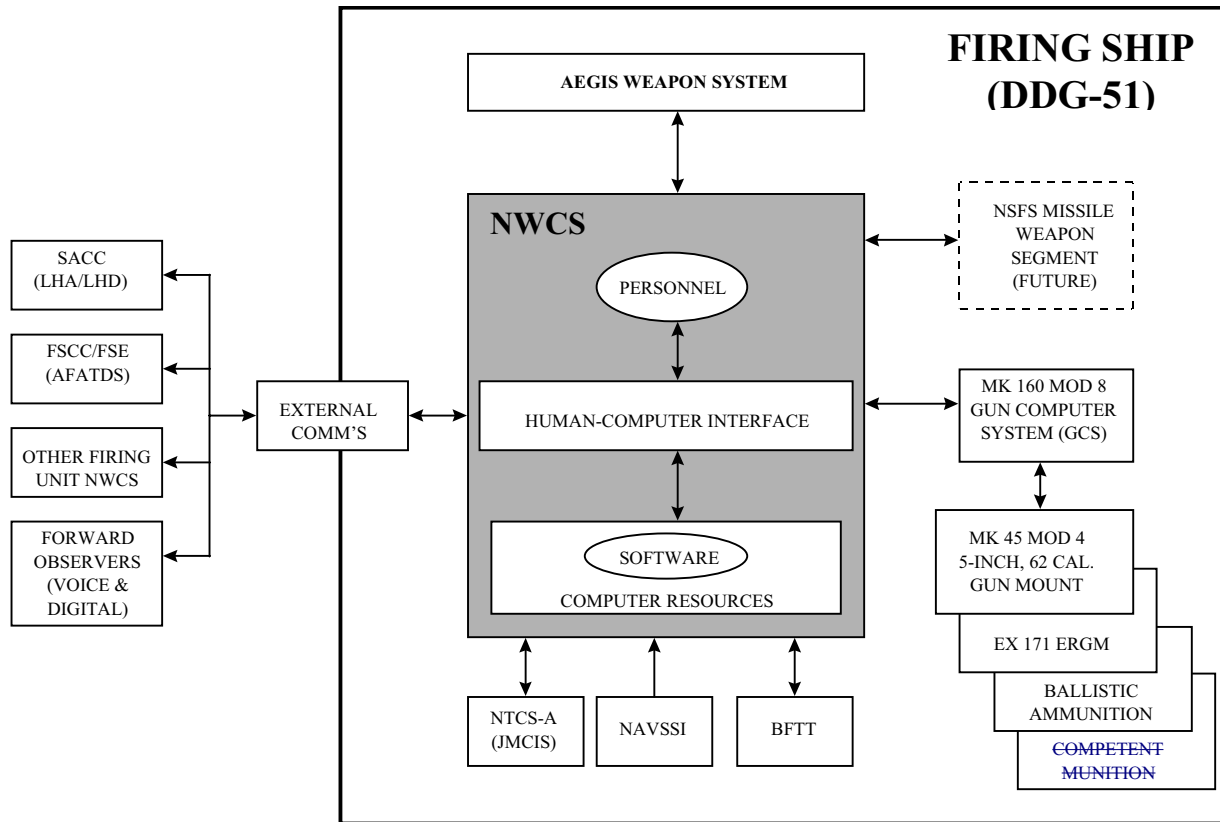


Figure E-3. Notional NFCS Component Architecture

The final step is the development of a high-level systems architecture for NFCS and the major systems that it interfaces with. [Figure E-4](#) is the current illustration of this high level NFCS environment architecture.

It is also valuable in an examination of the NFCS architecture to understand the communications requirements for the system. [Table E-1](#) and [Table E-2](#) summarize the message sequences and communications requirements for NFCS in Call-for-Fire sequences and general C4I coordination.

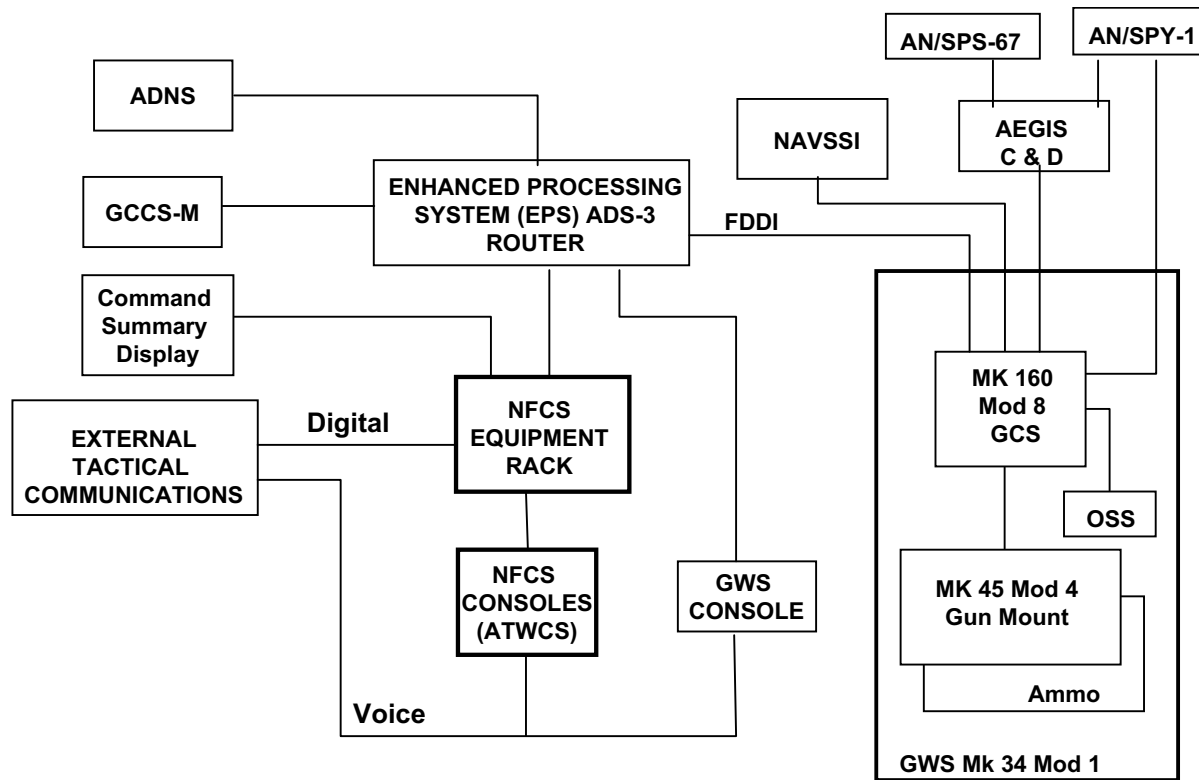


Figure E-4. Phase 1 NFCS Block Diagram

Table E-1. Call-for-Fire Message Sequence (Sheet 1 of 2)

Inputs to NFCS	NFCS Processing	Outputs from NFCS
GWS Fire Control Status GWS Gun Mount Status	Send status report to the FO. Sends identification, reaction time, weapon type, max/min range, munitions type, and other data.	Fire Unit Status (K02.18)
Observer Status (K02.50)	Store observer identification, location and effective time.	NFCS Spotter/Reference Point message
Fire Support Meteorological Data (K02.3)	Establish a meteorological database, consisting of Met station locations and altitude zones. As option, enter data manually.	NFCS Meteorological Data message

Table E-1. Call-for-Fire Message Sequence (Sheet 2 of 2)

Inputs to NFCS	NFCS Processing	Outputs from NFCS
Call-for-Fire (K02.4) GWS Target Status GWS Mission Status	Establish the Fire Mission Type, target location, target attributes file, mission time, ammunition type, and Method of Control. Order GWS to engage, evaluate fire control solution, Report “Ready.”	NFCS Target Assign message NFCS Engagement Data Message Observer Mission Update(K02.6)(K02.14)
GWS Trajectory message Call-for-Fire (K02.4) or On-call Fire Command (K02.12)	Evaluate the deconfliction constraints. Execute the first fire. If “At my Command,” then K02.12 executes.	NFCS Engagement Control Message
GWS Round Fired message	Send observer notification “Shot” and “splash” reports.	Message to Observer (K02.14)(K02.6)
Subsequent Adjust (K02.22)	Store and manage adjustment data.	NFCS Adjustment Data
On-call Fire Command (K02.12) or Call-for-Fire (K02.4)	Execute the adjust fire.	NFCS Engagement Control Message
GWS Round Fired message	Send observer notification “Shot” and “splash” reports.	Message to Observer (K02.14)
On-call Fire Command (K02.12) or Call-for-Fire (K02.4)	Execute the Fire for Effect.	NFCS Engagement Control Message
Target Data (K02.9)	Used for Record As Target operation, if desired.	NFCS Record As Target message
Check Fire (K02.1)	Order GWS to comply with FO commands.	NFCS Engagement Control Message
GWS Mission Status	When rounds complete, or mission duration expired, send Break Assign and Break Engage to GWS. Report end of mission.	End of Mission Notification (K02.25) NFCS Target Assign NFCS Engagement Control Message
End of Mission and Surveillance (K02.16)	When FO cancels mission, send Break Assign and Break Engage to GWS.	NFCS Target Assign NFCS Engagement Control Message

Table E-2. External Communications (VMF) C4I-Coordination

Inputs to NFCS	NFCS Processing	Outputs from NFCS
GWS Engagement Trajectory Data or GWS Trial Solution n	Obtain trajectory data. Specify target number, location, weapon type, ownship location, munitions type, fuze type, and maximum ordinate.	Request for Clearance to Fire (K02.21)
Mission Clearance (K02.13)	Receive approved/disapproved-approved in response to K02.21 disposition action (4063 001).	N/A
Fire Support Coordination Measures (K02.15)	Display appropriate lines, areas, zones, or boundaries, and airspace coordination areas, on tactical situation display. Maintain data for use in deconfliction process.	N/A
Call-for-Fire (K02.4)	NFCS send to notify of air warning area and effective time for a pending firing. Describes location and ceiling of the area.	In Progress Mission Notification (K02.24)

A Specific NFCS Interface Example: NFCS to GCCS-M Interface

One of the essential requirements for NFCS is that the system have an effective interface with Global Command and Control System—Maritime (GCCS-M) in order to receive information on tracks and track attributes, meteorological and oceanographic information, friendly forces disposition, and air tasking orders. This interface will consist of communications and database interaction functions in the Application Support Service area to ensure that messages and track information are effectively.

Interface Description

The GCCS-M Tactical Management System (TMS) a DII COE component which provides database management of tactical track data. (A track is a physical or imaginary object whose existence and, possibly, movement have been reported—i.e., tracked’’.). The GCCS-M track database consists of data that includes—but is not limited to—the identifying attributes and positional histories of reported ships, submarines, aircraft, land units, and other moving or fixed objects of interest.

This key interface between NFCS and GCCS-M will use a client-server application called the Track Database Manager (Tdbm) which is designed to operate within a local area network (LAN) of computer workstations using a virtual machine architecture. A Tdbm component is run on each machine in a network. One of the machines is designated the Tdbm Master Host. GCCS-M will contain the Tdbm master. Each other Tdbm Service is designated a client on the LAN. NFCS will host a client Tdbm.

The Tdbm process running on the Tdbm Master Host machine plays the role of network controller and synchronizes the track database management activities of each Tdbm within the network. Application code logs into the client Tdbm to make database changes and/or queries. The NFCS client Tdbm maintains sufficient data in memory to answer most application queries without accessing the Tdbm Master.

GCCS-M will also provide an interface for NFCS to receive certain data communications directly including Air Tasking Order (ATO) data and Air Coordination Order (ATO/ACO) data. This information will be received by NFCS in the form of a USMTF message via TCP/IP. GCCS-M will be set up to auto-forward the ATO/ACO messages to NFCS.

Results and Findings

The NFCS technical architecture model, illustrated in the following diagram ([Figure E-5](#)), is set up as a layered TRM structure. The TRM layers define boundaries that should be used when developing NFCS computer hardware and software interfaces. The TRM abstract model helps NFCS software designers and developers recognize interfaces and modularize the software into layers. These layers are used to pass information among other components and system.

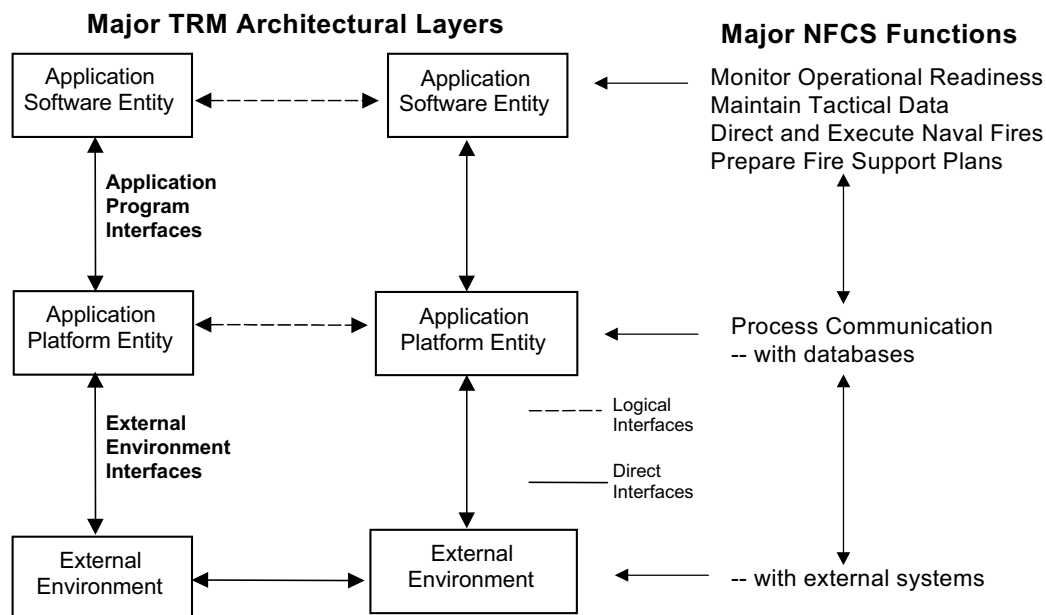


Figure E-5. NFCS Functions and Layered TRM Architecture

[Figure E-6](#) illustrates the use of the TRM services view to structure the services required for NFCS. The traditional TRM services view fits the NFCS application services layer well as it provides the primary communications and support applications services required to enable NFCS to function. Additionally, the model requires little additional work other than to identify the mission support applications (user applications) that NFCS requires. The TRM services view provides an extensive checklist against which to evaluate the completeness of the final NFCS design.

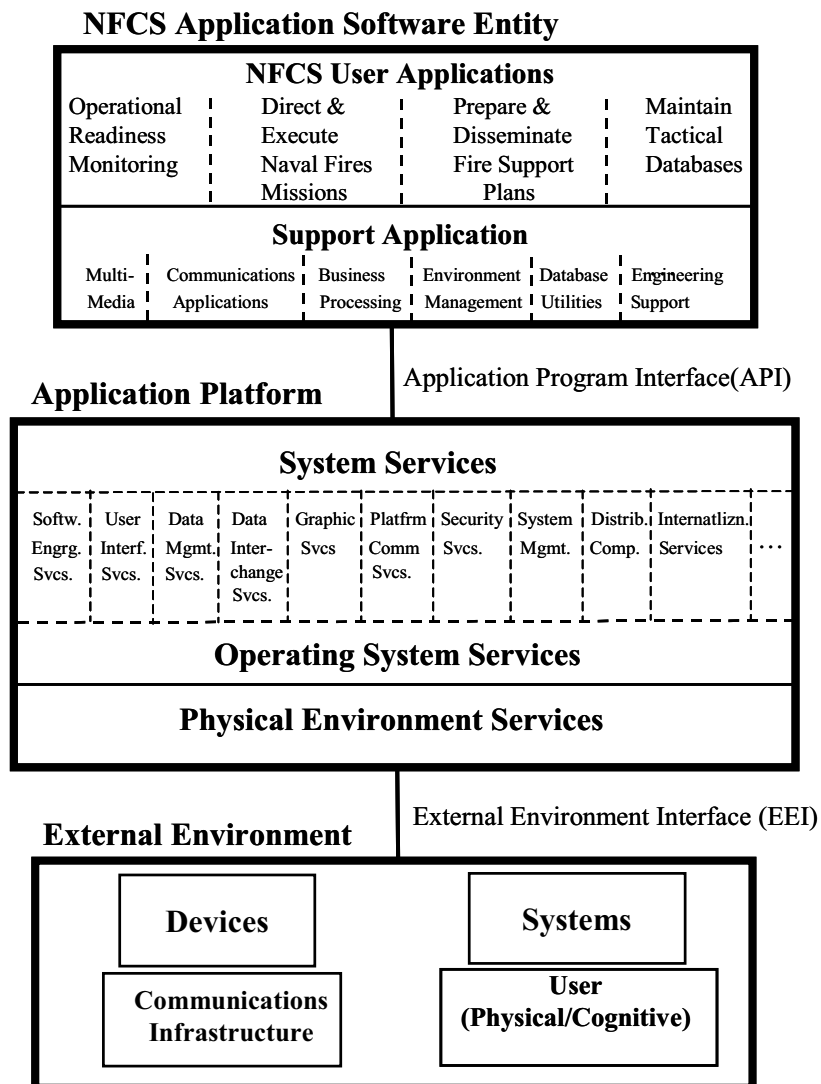


Figure E-6. NFCS Services Model

NFCS to GCCS-M Interface Mappings

An interface mapping between NFCS and GCCS-M indicates involvement in all of the service areas of the model. Generally, the Application Software Entity, where User Applications operate, would stimulate an exchange of information at the application layer via database utilities executing queries, retrieving data, and storing the results. [Figure E-7](#) illustrates the primary operating components of this interface.

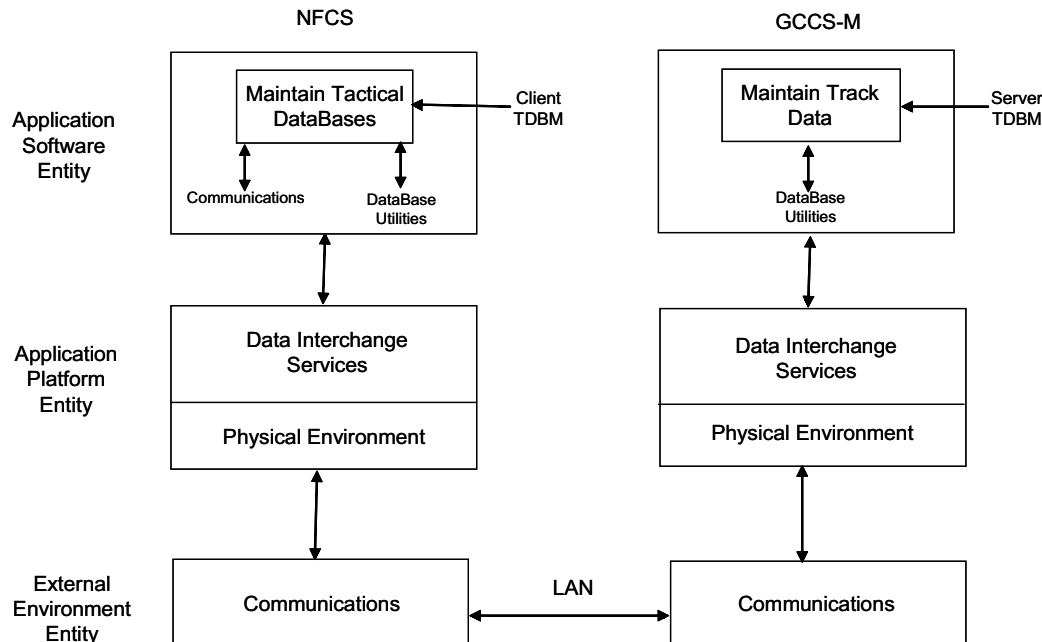


Figure E-7. NFCS to GCCS-M Track Data Interface

Conclusions

The primary conclusion to be drawn from an effort to map the NFCS architecture using the TRM is that the process is relatively straightforward and potentially useful in providing reference information for standards within the NFCS development effort. Contributing standards can be identified relatively easily once the interfaces have been mapped to the model. It is also believed that interoperability will be more assured considering the application of a systematic layering of functions, interfaces, services, and the use of standards.

NFCS architecture development efforts are in their early stages and significant work remains to be done. Additional effort must be undertaken to establish the applications required in more detail and to complete the mapping of services and interfaces to specific hardware, software and standards. NFCS architects are aware that significant challenges remain in establishing detailed information exchange and interface performance data.

References

1. Operational Requirements Document for a Naval Fires Control System (ACAT III), version 3, 11 June 1999
2. Surface Combatant Land Attack Guidance Document, 30 August 1999 draft
3. Naval Surface Fire Support Concept of Employment, version 3, March 2000
4. DoD Technical Reference Model (DoD TRM), Version 1.0, 5 November 1999
5. DoD Joint Technical Architecture, Version 3.0, 15 November 1999
6. NSFS Mission Need Statement, May 1992

Case Study 2: C4ISR Battle Group/Amphibious Ready Group (BG/ARG) Functional Design

SPAWAR

This case study describes the C4ISR Battle Group/Amphibious Ready Group (BG/ARG) Functional Design developed at the Space and Naval Warfare Systems Command. The purpose of the C4ISR BG/ARG Functional Design is to depict a notional BG/ARG functionality that can be used by Navy C4ISR resource sponsors, system planners and system implementers to help:

- Assessing future battle group functionality
- Planning efficient allocation of battle group system resources
- Tracing the operational requirements and supporting system capabilities
- Ensuring interoperability among and external to the battle group
- Identifying interoperability issues and system shortfalls

[Figure E-8](#) illustrates an overview of the BG/ARG Functional Design concept. The concept demonstrates a logical linkage from operational activities to system functions supported by interoperability standards. The Functional Design starts by identifying all the operational activities with associated information exchange requirements (IERs) that apply to the BG/ARG warfare command node(s). The operational activities are then linked to the system functions for subsequent allocation to the physical nodes. System function IERs are also developed, attributed, and documented. System functions are analyzed to derive applicable service areas and appropriate interoperability standards are defined for each service area. Accordingly, the BG/ARG Functional Design with three Architecture Framework views is developed: the Operational View (OV), the Systems View (SV) and the Technical View (TV).

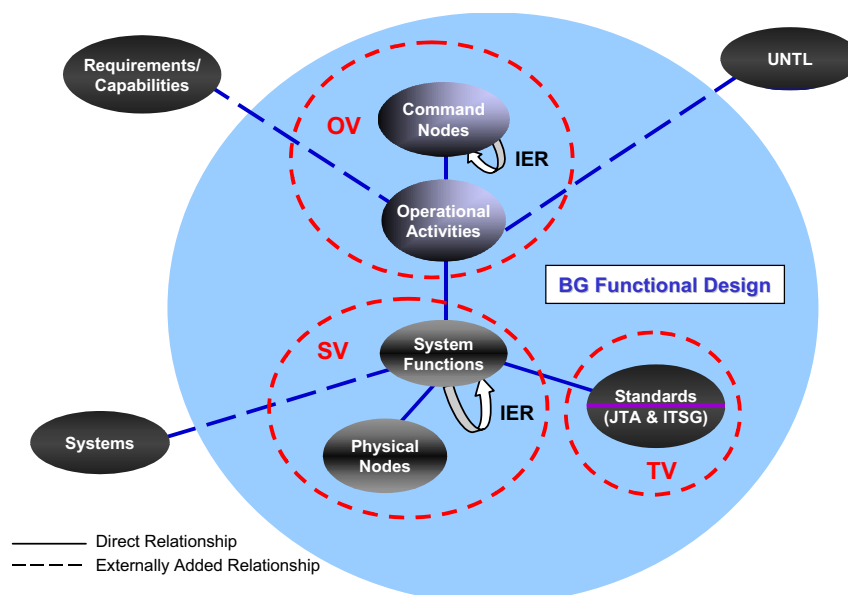


Figure E-8. BG/ARG Functional Design overview.

Rationale/Purpose

This case study demonstrates the usage of the DoD Technical Reference Model (TRM) as an aid in the development of the TV for the BG/ARG Functional Design. The DoD TRM was used as the reference for a list of all possible service areas that may be applicable to the BG/ARG Functional Design. The TV was developed by identification of DoD TRM service areas applicable to the BG/ARG system functions (from the SV). Standards were then selected for each service area based on review and analysis of the Joint Technical Architecture (JTA) and the DoN CIO Information Technology Standards Guidance (ITSG).

Description of Process Used

The process used for defining the BG/ARG Technical View (TV) is as follows:

1. Review the DoD TRM System Services for possible application to the BG/ARG Functional Design
2. Determine applicable TRM System Services that apply to the BG/ARG system functions
3. Select standards to pertinent System Services

Service Areas for the BG/ARG TV were derived from Service View in DoD TRM dated 5 Nov 99 and were amplified for application to the BG/ARG Functional Design. The BG/ARG service areas are mapped to the Systems Services in the DoD TRM as follows ([Table E-3](#)):

Table E-3. DoD TRM System Services to BG/ARG Service Areas (Sheet 1 of 2)

BG/ARG ServiceAreas	DoD TRM System Services
Document Interchange Graphics Interchange Geospatial Interchange Still imagery Interchange Motion Imagery Interchange Audio Interchange Atmospheric Interchange Oceanographic Interchange Signal Intelligence Imagery Intelligence	Data Interchange Services (DoD TRM Section 4.4.2.1.4)
E-Mail VTC Network Services Transmission Media	Communication Services (DoD TRM Section 4.4.2.1.6)
Information Modeling	Engineering Support (DoD TRM Section 4.4.1.2.6)

Table E-3. DoD TRM System Services to BG/ARG Service Areas (Sheet 2 of 2)

BG/ARG ServiceAreas	DoD TRM System Services
Data Management	Data Management Services (DoD TRM Section 4.4.2.1.3)
Human Computer Interface	User Interface Services (DoD TRM Section 4.4.2.1.2)
Security	Security Services (DoD TRM Section 4.4.2.1.7)

Results/Findings

Using the detailed descriptions of the System Services in the 5 Nov 1999 TRM document, suitable service areas (listed above in [Table E-3](#)) can be easily synthesized. These service areas provide a framework for interfaces and subsequent analysis of interoperability issues. There are a number of useful applications that can be derived from the BG/ARG function design by using the design elements (e.g., operational activities, system functions, standards, physical nodes) shown in [Figure E-8](#).

- **Interoperability Issues**—Identify any difference in standards or performance parameters that jeopardizes information exchange among command nodes. Ensure that standards are compliant with mandated and governing documents such as the JTA and ITSG/ITIA. The set of service areas derived from the DoD TRM provides a common framework for selection of interfaces and standards.
- **System Tradeoff Threads**—Identify links between system functions and systems; compare systems providing similar functions.
- **Functional Gaps and Overlaps**—Analyze system functions to determine functional redundancies and/or deficiencies.
- **Personnel Loading**—Analyze system functions to determine personnel requirements and further assess overlapping efforts.

Expectations

The DoD TRM is useful in providing insight into the selection of service areas applicable to the BG/ARG functional design. The set of service areas derived from the DoD TRM provides a common framework for selection of interfaces and standards. Analysis of these standards and interfaces across a multitude of command nodes enables identification of interoperability issues that need to be resolved.

References

Relevant reference material used to develop the Technical View of the BG/ARG Design Document comprises the documents identified herein.

1. DoD Technical Reference Model (DoD-TRM), Version 1.0, 5 November 1999
2. SPAWAR C4ISR BG/ARG Functional Design, Version 0.2, 31 January 2000
3. C4ISR Architecture Framework, Version 2.0, dated 18 December 1997
4. Department of Defense Joint Technical Architecture, Version 3.0, dated 15 November 1999
5. DoN Information Technology Standards Guidance (ITSG), Version 99-1, dated 5 April 1999
6. Department of the Navy Information Technology Infrastructure Architecture, Version 1.0 (Proposed), dated 16 March 1999
7. OPNAVINST 3500.38, Universal Naval Task List (UNTL), Version 2.0, Draft #7, August 1999
8. Strike Warfare C4ISR Operational Architecture and Preliminary Improvement Analysis (STWOA) Report, June 1997

Case Study 3: Transmission Control Protocol/Internet Protocol (TCP/IP) to TRM

**US ARMY Tank-Automotive Research,
Development and Engineering Center (TARDEC)**

Objective of Case Study

This case study will show how the Transmission Control Protocol (TCP) Internet Protocol (IP) communication suite of protocols uses a layering implementation that can be modeled via the DoD Technical Reference Model which uses defined direct and logical interfaces. This case study provides a very general description of how TCP/IP relates to the TRM, and it doesn't provide an in-depth explanation of TCP/IP.

Rationale/Purpose

The Technical Reference Model (TRM) is an evolution between the TAFIM and the GOA model. The TAFIM view defines services within layers, and the General Open Architecture (GOA) standard, SAE AS4893, to define the interfaces between layers. These layers share information with each other through a defined set of protocols. The TRM stack (stacked layers) begins with the physical hardware layer and proceeds through a set of layers to the application layer via a precise set of direct and logical interface definitions that clearly identify the separation of boundaries between the layers. A layered structure provides the following attributes: layer portability, interoperability between associated layers, plug and play implementations, and affordability via contractor competitive competition for the development of each of the independent layers.

The purpose of this case study is to show that the TCP/IP suite of communication protocols can be modeled using the TRM.

Description/Process Used

In the TRM to TCP/IP diagram ([Figure E-9](#)), the model is setup as a layered structure. The TRM layers define boundaries that should be used when developing computer hardware and software interfaces. The TRM abstract model helps a software developer recognize interfaces and modularize the software into layers. These layers then pass information amongst other components and systems.

To develop a communication suite of protocols that are portable and easily maintain, the TRM is the modeling tool to help identify the hardware/software layers needed to pass information. A concern in passing information is how to identify and define the data parameters needed to transfer the information from one platform to another platform successfully keeping in mind the separation of boundaries to the input layer, output for that layer, and the logical interface understanding between adjacent components or systems. A software developer must also look at how the passed information will be protected and verified correct upon receipt. Header data, bit

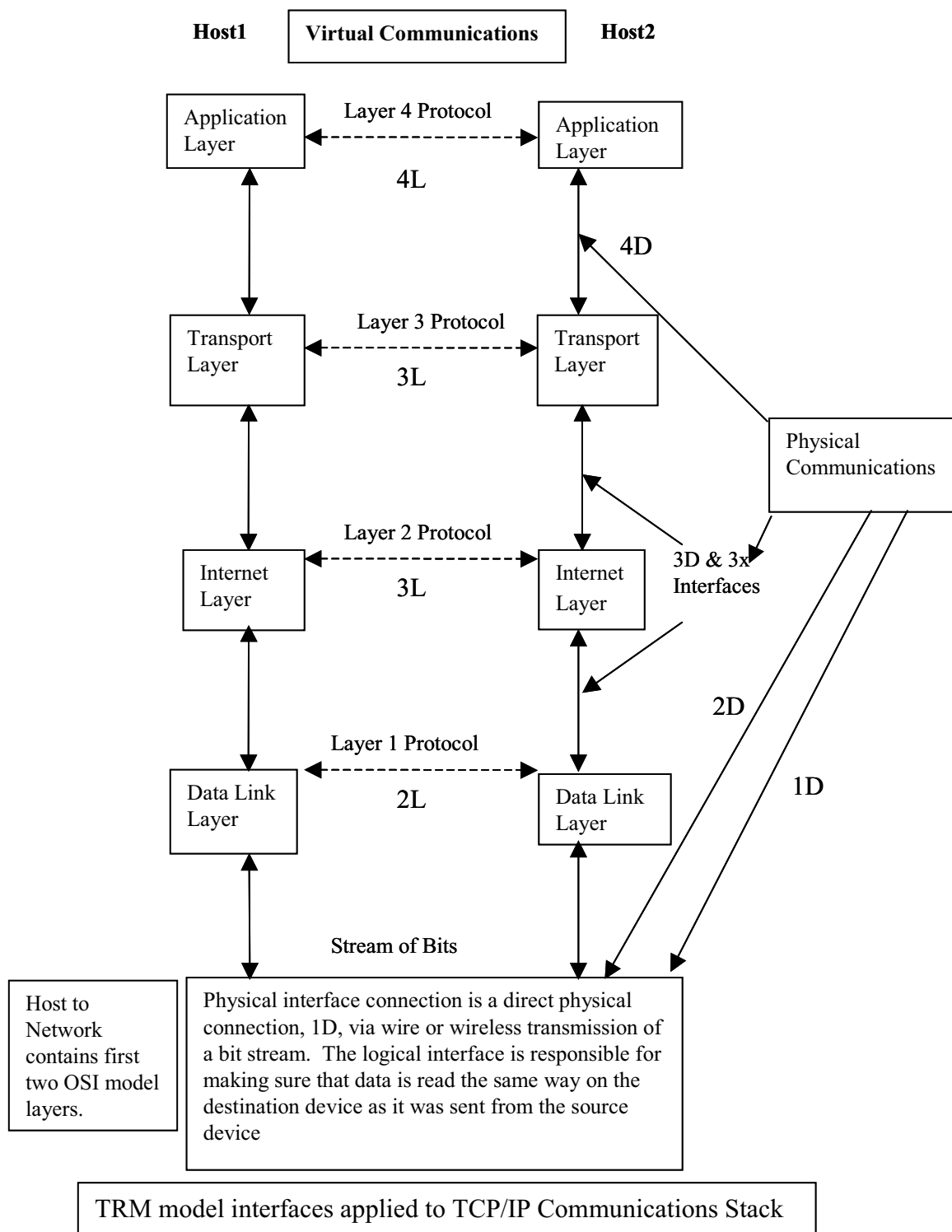


Figure E-9. TRM Model Interfaces Applied to TCP/IP Communications Stack

insertion techniques, and various algorithms are used to ensure application data within a bit stream is not corrupted. Also within a communications protocol, the receiving layers should send acknowledgements back to the sender of the information letting that layer know that the passed information was successful. If not successfully passes, the sending layer will retransmit the information.

An example of such a communications protocol is the Transmission Control Protocol (TCP) Internet Protocol (IP). TCP/IP uses four stacks or layers to transfer data, because the physical layer and data link layer are combined. In the diagram, I have broken these two layers out to show the distinction between the layers.

The purpose of a layer is to offer certain services to the higher layers, shielding those layers from the details of how the services are actually implemented. Layer n on one-machine carries on a conversation with layer n on another machine via the protocol definition (logical interface definition). A resulting communications protocol stack is formed when data is passed from each individual layer from the lowest layers to the highest layers and the highest layers to the lowest layers. [Figure E-9](#) shows a protocol stack formation, and the communications between host1 to host2.

The virtual communications between host1 and host2 is the logical interface classes, which establishes the communications understanding or protocol understanding between layers. The physical communications is the direct software layer algorithm to direct software layer algorithm interface, which passes and receives the defined software data parameters based on the logical interface definition. The software layer algorithms are created based on the protocol definition, and each software layer algorithm implementations are independent from other software layer algorithms. Virtual communication (logical interface) would not exist without the direct interfaces, and the logical interface definition determines the data parameters that the direct interface algorithms utilize.

Below is a description of how the TCP/IP packets (stream of data) of information that are separated into columns of information which form the layers in the TCP/IP stack.

In the TCP/IP communication protocol, the architecture is based on the Internet datagram definition ([Figure E-10](#)). The make up of the datagram is a stream of bits, 1's and 0's. This stream of bits is further defined by bytes. There are 8 data bits to one byte. The Internet datagram is defined as:

Physical/Data Link	IP Layer Data	TCP or UDP Layer	Application Data	Frame Check Sequence
--------------------	---------------	------------------	------------------	----------------------

Figure E-10. Internet Datagram Definition

Note: The Frame Check Sequence is used as error checking in data transmission and UDP is used for error checking when send isolated messages to another system.

A datagram is a finite-length packet with sufficient information to be independently routed from source to destination without reliance on previous transmissions. Each of the separately defined byte lengths represent data that the software layer algorithm understands via the protocol definition.

Physical Layer

The TRM physical layer is responsible for generating and actually passing the physical electrons, stream of information, from sending hardware component to the receiving hardware component. The physical layer contains the hardware, sensors, microcontroller, microprocessors, data buses, electromagnetic waves, and electrical interface requirements, which is the 1D direct interface.

- Logical Interface

An example of the logical interfaces associated with the physical layer bit stream of information being passed from the sending hardware to the receiving hardware is shown below ([Figure E-11](#)).

```
011111 11011111 010 Data Bits
0111110110111110010 Stream of Information being Transferred
```

Figure E-11.

Extra Bits Inserted and Removed by the Hardware

The sending hardware inserts a zero bit (0) after every five consecutive one-bit pattern (1). This procedure is called zero bit insertion or bit stuffing. The receiving hardware will remove the inserted zero bits upon receipt of the data stream. The two hardware components must have already defined the logical interface definition before bit insertion. So, the sending and receiving hardware already have an understanding on how they will communicate with each other before sending and receiving information data. This logical interface definition is normally used so duplication for the user data (application data) is not corrupted from sending hardware to receiving hardware.

- Direct Interface

The direct interface is a bus connecting the two hardware pieces.

Data Link Layer

The task of the data link layer is to convert the raw bit stream offered by the physical layer into a stream of frames for use by the network layer. Various framing methods are used, including character count, character stuffing, and bit stuffing. Data link protocols can provide error control to retransmit damaged or lost frames.

Internet Protocol Layer

The Internet Protocol (IP) layer routes data between the hosts, other systems. The information data may be passed to a single network or may be relayed across several networks in the internet. IP address data routes its traffic without caring which application to application interaction a particular datagram belongs to.

Transmission Control Protocol (TCP)

The TCP provides a reliable data connection service to applications. TCP has as an algorithm, which guarantee that data is error free, complete, and in proper sequence for recombining packets of data. TCP ensures data is sent and received by sending acknowledgement message between the sending TCP layer system and receiving TCP layer system. It is a logical interface or virtual communications interface.

Application Layer

Application used by TCP/IP are: File Transfer Protocol, Simple Mail Transfer Protocol, Telnet terminal access, Domain Name System (DSN) directory services, and program to program communications.

[Figure E-12](#) and [Figure E-13](#) show how the information data can be separated into segments and represented as a layers/stacks.

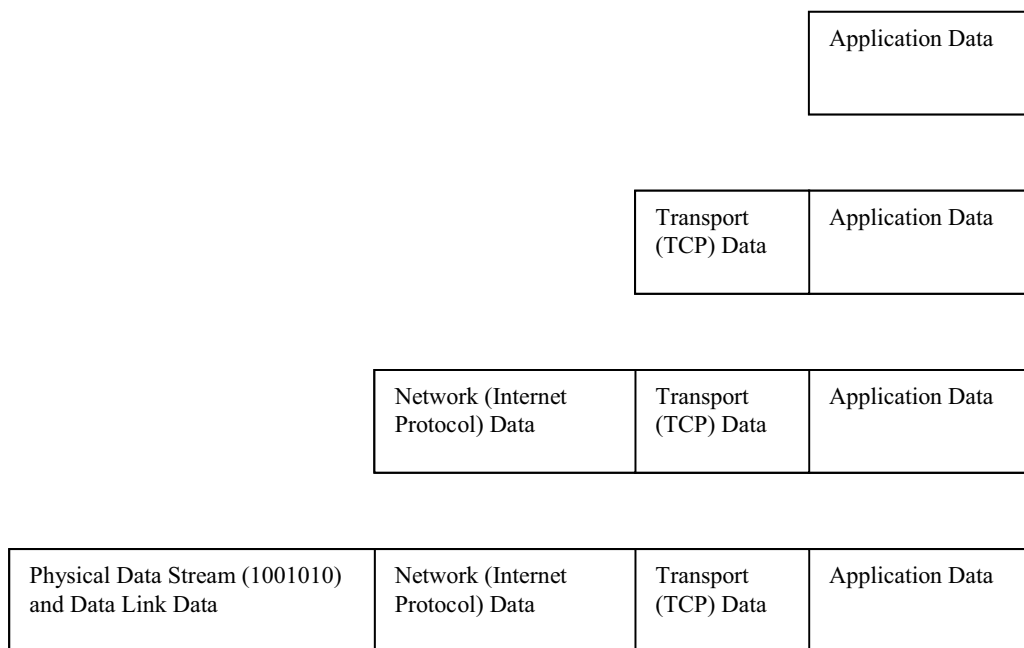


Figure E-12.

Relationship between OSI and TCP/IP Models

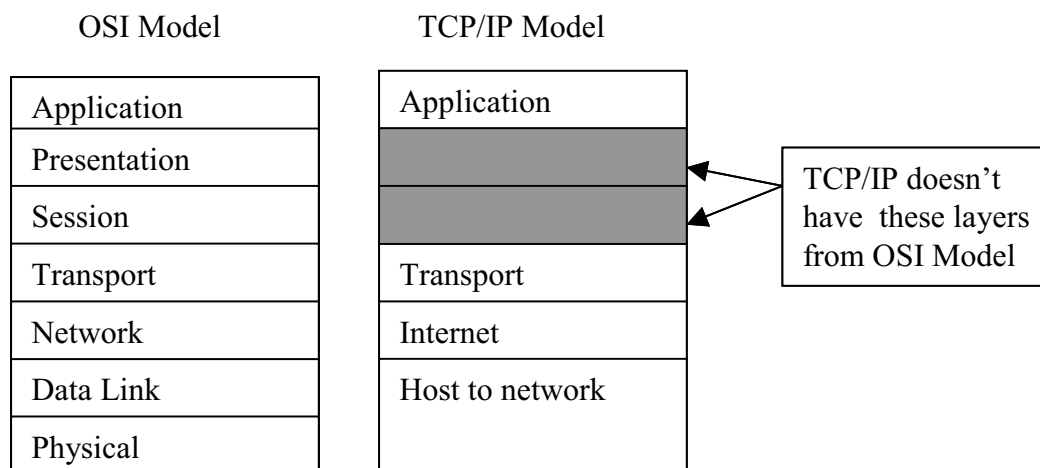


Figure E-13.

Results/Findings

In the comparison between the TRM, OSI, GOA and TCP/IP, I have broken down the TCP/IP model using the TRM and OSI model concepts. The TCP/IP model doesn't make a distinction between the physical and data link layers. Each layer in the TCP/IP model uses the fields in the datagram above to check for error transmissions and router destinations while protecting the application data. Each of the defined fields, bytes, are stripped away via a particular layer algorithm as defined by the protocol being used. When the remaining datagram bit stream reaches the application layer, the bit stream contains some header information for error checking and the application data. The application software can use the application data without having any knowledge of how the data arrived.

[Figure E-14](#) is another view of the relationship between the TRM model, TCP/IP and the GOA interface classes. The TCP/IP model can easily be represented via TRM model and OSI model.

Definitions

Destination Address: This field contains the destination information (used by the network Layer).

Source Address: This field contains source information (used by the network Layer).

User Data field: This field contains data used by the application (used by application).

Frame Check Sequence: Error checking for data transmission

Service is a set of primitives (operations) that a layer provides to the layer above it. The service defines what operations the layer is prepared to perform on behalf of its users, but it says nothing at all about how these operations are implemented.

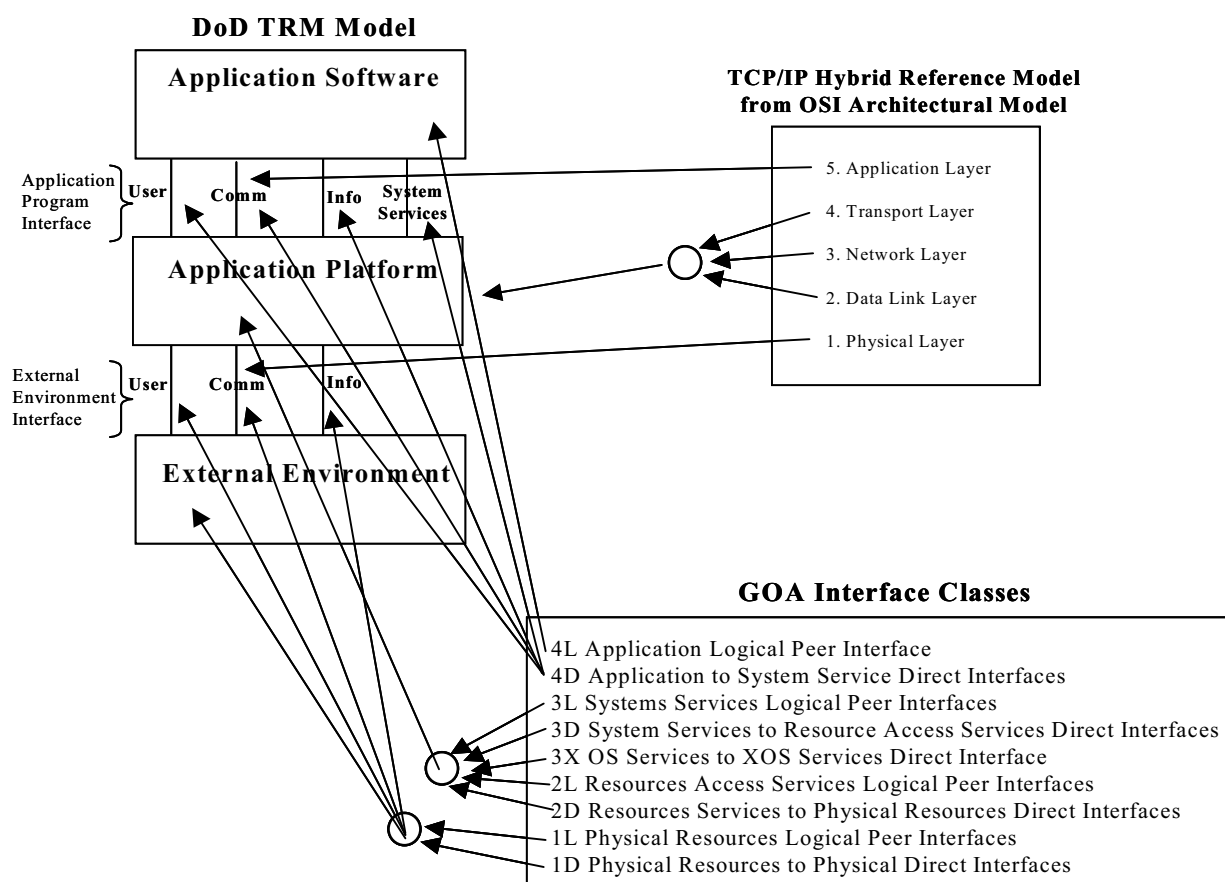


Figure E-14.

Protocol is a set of rules governing the format and meaning of the frames, packets, or messages that are exchanged by the peer entities within a layer. Layers use protocols in order to implement their service definitions.

Host to Network (first two layers):

- The physical layer is concerned with the transmission characters of wire, fiber optics, and wireless communications. It passes the stream of bits in the form of a datagram.
- The data link layer delimits the start and end of frames to be used by the network layer.

The network layer is concerned with getting packets from the source to the destination.

The transport layer provides reliable, cost-effective data transport from the source machine to the destination machine, independent of the physical network.

The application layer is concern with security, naming within the internet, network management, and the applications such as electronic mail, net news, multimedia, and world wide web.

Expectations

The TCP/IP communications protocol is an example of how to define, using the TRM model, interfaces within an system. Defining the interfaces between layers makes the software portable. A common understanding (datagram definition, [Figure E-10](#)) should be established between system layers for the purpose of interoperability. The TRM model provides the layer concept needed to support interoperability, portability, and affordability. The TRM model is a good tool to use for computer and software developers.

Main domains use the TCP/IP in their communications with each other. (DoD, Industry, Various Countries). Interoperability between these domains is achieved because they use the same standard (TCP/IP standard). JTA accomplishes the same interoperability goals mandating standards that are to be used by the weapons community.

References

1. TCP/IP (Architecture, Protocols and Implementation) by Sidnie Feit
2. Computer Networks, Third Edition, by Andrew S. Tanenbaum
3. DoD TRM Version 1.0, November 5, 1999
4. SAE AS4893

Case Study 4: Mapping of Defense Information Infrastructure (DII) Common Operating Environment (COE) Segments to DoD Technical Reference Model (TRM) Services The Aerospace Corporation

Purpose and Objective of the Case Study

The Defense Information Infrastructure (DII) Common Operating Environment (COE) is an infrastructure for building interoperable systems across user applications using a set of guidelines, standards and specifications implemented through a collection of reusable components or segments. In the DII COE Integration and Runtime Specification (I&RTS), a segment is defined as a collection of one or more software and/or data units most conveniently managed as a unit of functionality. Building a target system includes combining COE components with mission-specific software.

The mapping document categorizes the DII COE segments into the service areas defined in the DoD TRM. Each COE segment is mapped into a DoD TRM service based on the segment's functionality as determined from DII COE and COTS documents. Although the DoD TRM provides for both a services and an interfaces view, at this time the mapping contains only the services view.

Rationale

In the DII COE documentation, a conceptual correspondence already existed between the DoD TRM and the DII COE, illustrated in figures showing the DII COE services and architecture. As a logical follow-on to the notional mapping already in the DII COE documentation, a finer granularity of mapping was developed of the DII COE segments to the services defined in the DoD TRM. The mapping provides a common foundation for viewing diverse components from a conceptual perspective, and helps to understand the roles of the various COE segments. It can be used not only to contrast DII COE with other system architectures but also to provide a basis for correlating interoperability areas between the DII COE and other system architectures.

Description

The COE segments referenced in the mapping were derived from the DII COE 3.4 Build Lists. The purpose of each segment in the build list was identified, and based on the segment's purpose, it was mapped to a DoD TRM service. The software version description document and the DII COE Integration and Runtime Specification (I&RTS) document were used to obtain a segment's function. If a software version description document was unavailable for a segment and the I&RTS provided no descriptive information, the World Wide Web was searched for information related to the segment.

The mapping document contains a listing of the DoD TRM service areas, each followed by the descriptions from Section 4.4 of the DoD TRM. Below each DoD TRM service are the COE segments mapped to that service. With each segment is a description of the segment, with the source for the description identified. The host platform(s) on which each segment is available are also listed.

Results/Findings

Most of the DII COE segments were easily mapped to a single DoD TRM service area, but some of the segments could be categorized into more than one service area. To minimize ambiguity, the primary purpose of the segment was identified and used to categorize it into a single DoD TRM service area.

For example, the DocViewer segment allows users to view documents installed locally or documents loaded on any document server. This segment was mapped to DoD TRM 4.4.1.2.1 Multimedia.

Similarly, the Application Framework segment is a client to the Joint Mapping Toolkit—Visualization segment that provides a framework for application segments that share a common tactical display. The DII Motif Style Segment tailors and extends basic Motif features in accordance with Version 3.0 of the DII COE User Interface Specification. Both of these segments were mapped to DoD TRM 4.4.2.1.2 User Interface Services.

As another example, the Object Request Broker is an intermediary that coordinates and manages the requests between clients and servers. This segment was mapped to DoD TRM 4.4.2.4.4 Distributed Computing Services.

There were some segments that did not appear to map directly to an existing service area. For each of those segments, a decision was made to map the segment into the DoD TRM service area that seemed most closely related to the purpose of the segment. The segments where this tailoring of the model was required included the Software Development Kits (SDK), the Alerts, and the data segments.

The SDKs provide examples and libraries that a developer can utilize to develop or expand the function of a COE segment. Each SDK was categorized into one of the DoD TRM service areas based on the functionality developed by utilizing the SDKs. For instance, the Universal Communications Processor (UCP) SDK provides the development libraries, including scripts, data, and sample sources, for third party UCP developers to build client applications for the UCP engines. Since the UCP mapped to the DoD TRM 4.4.2.1.6 Communication Services, the UCP SDK was also mapped to that service area.

The Alerts segment provides a generic mechanism for the sending and receiving of alert messages between processes. It provides applications with the ability to register specific events which will generate visual/audio/log alerts to the operator during the operation of the system. This segment was mapped to the DoD TRM 4.4.2.2.9 Extended Operating Services, since the description of the services in that service area was closest to the functionality of the alerts.

The data segments can be classified into account groups, templates, and data used by runtime software. They don't provide any functionality by themselves, but rather provide a mechanism for creating or modifying the runtime environment. The account groups, which provide samples for accessing and customizing the user interface, were mapped to the User Interface Services. The templates provide examples of how to customize the behavior of a parent segment and were mapped to the same DoD TRM service area as the parent segment. The data segments were mapped to the same DoD TRM service area as the segment that utilized the data at runtime. For example, the MIL-STD-2525 Symbology data segment, which contains Computer Graphics Metafiles (CGM) and a menu hierarchy that allow the construction of MIL-STD-2525A icons, was mapped to DoD TRM 4.4.1.2.1 Multimedia.

Conclusions

The DoD TRM is very useful in gaining a better understanding of the roles and functions of the DII COE segments. Moreover, the mapping process also provides useful feedback on the DoD TRM itself. The mapping provided valuable insight into the tailoring process, and may even provide input in possible future updates of the DoD TRM.

References

1. Defense Information Infrastructure (DII) Common Operating Environment (COE) Integration and Runtime Specification (I&RTS), Version 3.1, Defense Information Systems Agency, 1 October 1998
2. Department of Defense Technical Reference Model (DoD TRM) Version 1.1 Coordination Draft, Defense Information Systems Agency, 15 July 1999
3. Mapping of Defense Information Infrastructure (DII) Common Operating Environment (COE) Segments to DoD Technical Reference Model (TRM) Services, Aerospace Report No. ATR-99(3583)-1, The Aerospace Corporation, 26 August 1999

Case Study 5: VETRONICS Technology Testbed (VTT)

US ARMY Tank-Automotive Research, Development and Engineering Center (TARDEC)

Objective of Case Study

The Vehicle Electronics (VETRONICS) organization within TARDEC developed the VETRONICS Open System Architecture (VOSA) model, which is the VETRONICS implementation of the Department of Defense (DoD) Technical Reference Model (TRM) concept.

The goal of VOSA is to address open system architecture concerns for ground platform systems using both commercial and military standards and to incorporate the standards identified by VOSA into the Joint Technical Architecture – Army document. VOSA has identified standard software/hardware interfaces, which facilitate interoperability; technology upgrades; reduced software development time; maximizes portability; and coordinates the sharing of resources.

This case study will show how TARDEC developed, as a research and development project, a new weapon system concept, the VETRONICS Technology Testbed (VTT), using the DoD TRM as the basis for the architectural design. VOSA via DoD TRM use interface classes to define the relationship between entities within a weapon system, and VETRONICS used VOSA to define the entity relationships within the VTT.

System Overview

The VTT is a research and development project within TARDEC being sponsored by the Office of the Deputy Assistant Secretary for Research and Technology, Dr. A. Michael Andrews. The VTT requirements were generated from the Office of the Deputy Assistant Secretary for Research and Technology and the user (soldier). The VTT ([Figure E-15](#)) is a ruggedized, real-time hardware and software module system being integrated into an actual ground combat host vehicle. The main objective of the VTT is to demonstrate the capability of one crewmember to perform the functions of both the vehicle Commander and vehicle Driver. The VTT will also be one of the beta test sites for the Weapon Systems Technical Architecture (WSTA) Operating Environment (OE). The demonstration must take place while operating over military significant terrain and while performing a military significant mission.

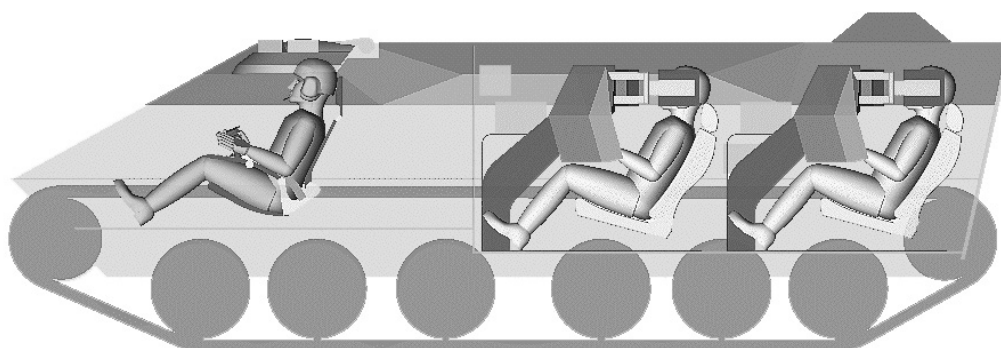


Figure E-15. VETRONICS Technology Testbed (VTT)

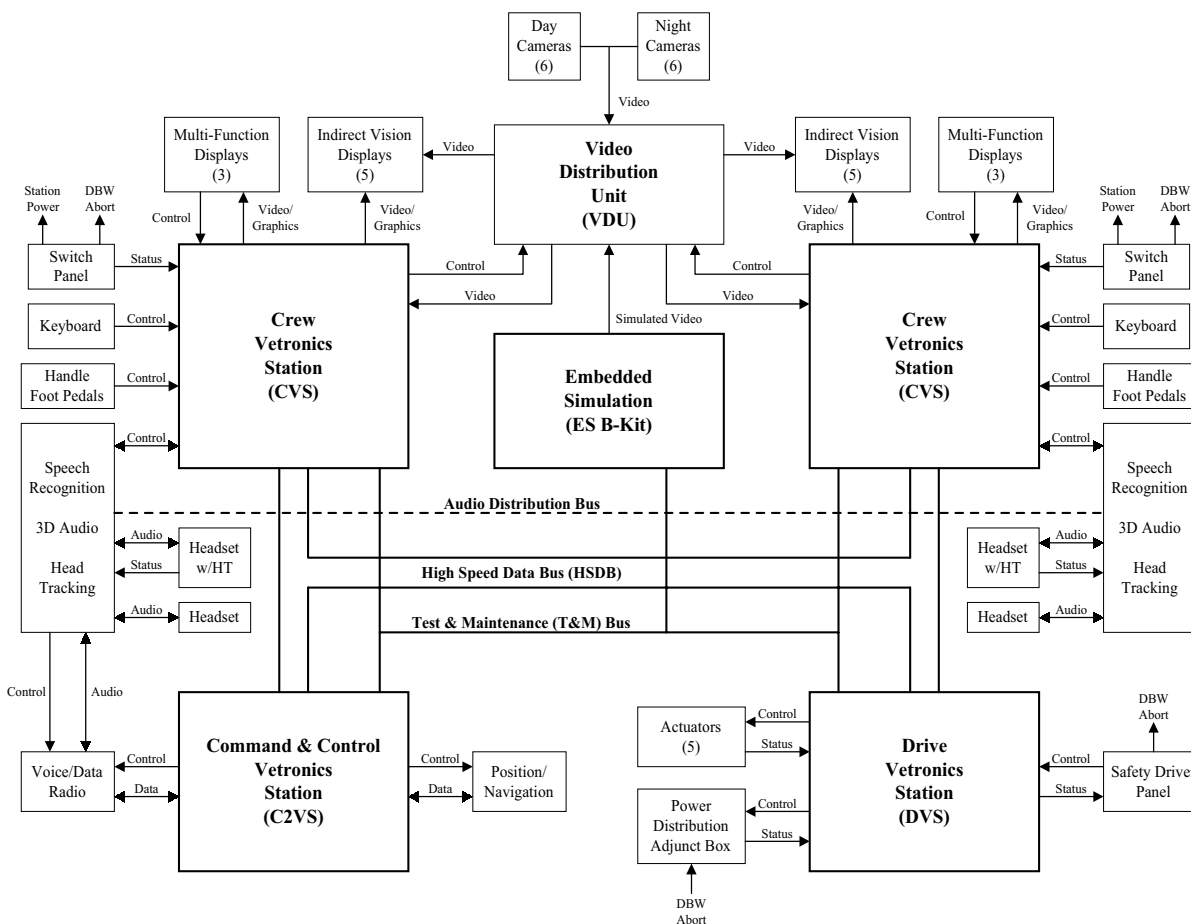
Vehicle requirements will include:

- Tracked Vehicle
- 20–40 Ton Weight Class
- Two Crewmembers (Commander/Driver and Gunner)
- Drive-By-Wire Capability
- Day and Night Operation
- Indirect Vision as Primary Vision
- Three Dimensional (3D) Audio System
- Speech Recognition and Generation
- Head Tracking
- Multi-Function Displays with Touch Screens
- Baseline Crewman's Associate Soldier Machine Interface (SMI)
- Embedded Simulation as an Enabling Technology for Embedded Training, Mission Rehearsal, Battlefield Visualization and After Action Review
- Compatibility with Joint Technical Architecture (JTA)-Army and Weapon System Technical Architecture Working Group (WSTAWG) mandates
- WSTAWG Operating Environment (OE)
- FBCB2 Interoperability (Embedded Battle Command (EBC) and Joint Variable Message Format (JVMF))
- Communication via Voice Intercom and SINCGARS Radios
- Data Logging/Test Observation is Required
- Vehicle Safety Release/Safety Driver

Description/Process Used

VOSA defines the open system standards needed for the ground combat weapon system. The VOSA architecture process is an iterative process involving studies of legacy systems, industrial standards and commercial standards that meet the Technical Architecture and performance requirements for the combat weapon system. This architectural iterative process involves studying cost factors, performance risks, and performance with respect to interoperability. The VTT is being simulated in a laboratory environment for preliminary testing, and the software used in this simulation testing will be installed on the vehicle for ground vehicle testing. VETRONICS is achieving software reuse by porting software from the Systems Integration Laboratory (SIL) to the combat vehicle.

System Level Architecture. The VETRONICS system of a combat vehicle ([Figure E-16](#)) shall consist of stations interconnected by data, audio, and a video distributed network all powered by a distributed power network. The components that make up the software architecture ([Figure E-17](#)) are identified by name with data flow direction via data buses identified in [Table E-4](#). An example of the software data type definitions used that provide for software interoperability between layers and plug and play capabilities are shown in [Table E-5](#).



Revision 11 - 14 February 2000

Figure E-16. VETRONICS System of a Combat Vehicle

To help us refine the architecture for the VTT, we identified the services and the interface classes within our standards using the DoD TRM guidance document ([Table E-4](#)). By using this table, the system interfaces and services can be categorized, assessed and compared for commonality, portability, and interoperability issues.

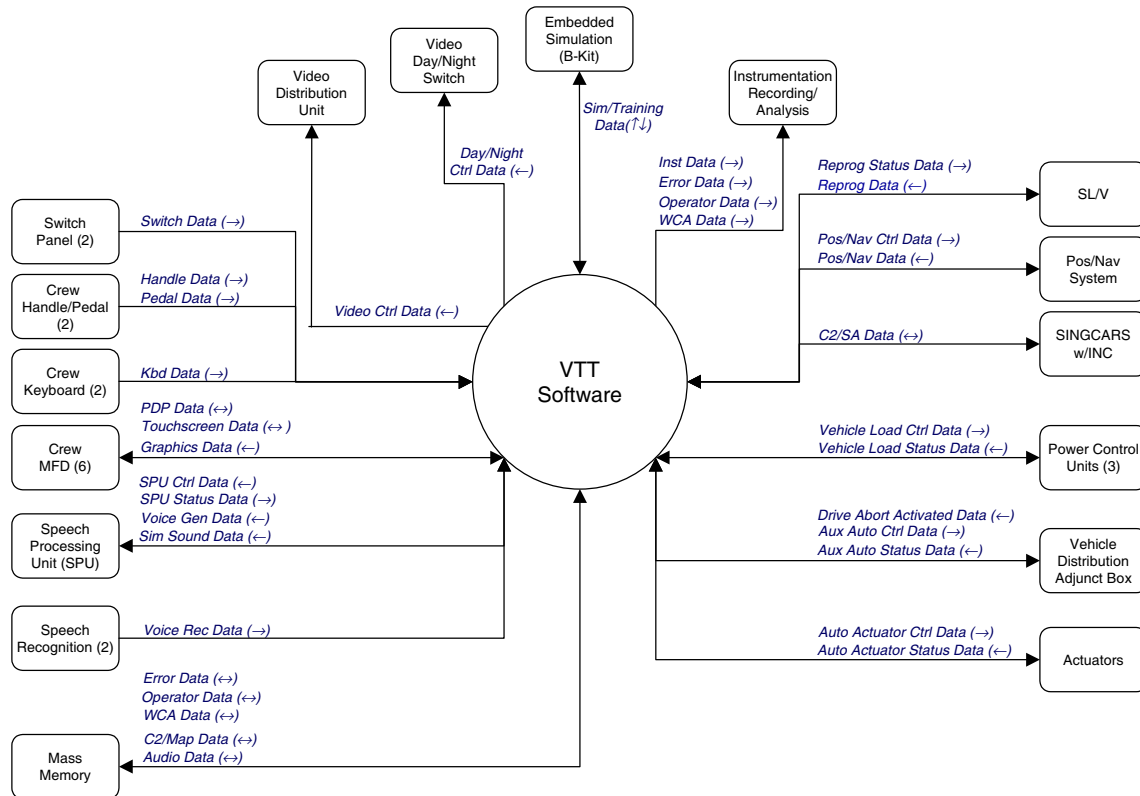


Figure E-17. Software Architecture

Table E-4. (Sheet 1 of 4)

VETRONICS TECHNOLOGIES TESTBED March 2000			Joint Technical Architecture (JTA) Version 3.0 15 Nov 1999
(Draft) Interface Definition	Standard	Description	Paragraph
1D	EIA 232	Serial Data Link	2.3.2.2.2.2
1D	EIA 423	Serial Data Link	2.3.2.2.2.2
1D	EIA 422	Serial Data Link	2.3.2.2.2.2
1D	MIL-STD-1275	Characteristics of 28 Volt DC Electrical Systems in Military Vehicles	
1D	MIL-STD-1474	Noise Limits for Army Materiel	
1D	MIL-STD-461	Requirements for the Control of Electromagnetic Interference Emissions and Susceptibility	
1D/1L	SVME-179	General Purpose Processors (GPP) PowerPC	
1D/1L	IEEE 802.3/80214	CSMA/CD, 10/100BASE-T	2.3.2.2.2.1
1D/1L		ATM: Asynchronous Transfer Mode	2.3.2.2.2.5
1D/1L	Dual Pentium III 500mhz	Processors B-Kit	
1D/1L	RT-1523(E)	Receiver-Transmitter (ASIP)	
1D/1L	AS-3916	Antenna	
1D/1L	IEEE 1014 Rev D	VME64 (VME Extension)	WS.GV.3.5.2
1D/1L	ANSI X3.131	Small Computer Serial (System) Interface	WS.GV.3.5.2
1D/1L	MIL-STD-1553B	Digital Time Division Command/Response Multiplex	WS.GV.3.5.2
1D/1L	IEEE P1386-1	PCI Mezzanine	WS.3.5.3
1D/1L	EIA 170	RS-170 Monochrome Video	
1D/1L	SMPTE 170M	NTSC Color Video	
1D/1L	IEEE 1102.2 (IEEE 1101.2)	Mechanical core specifications for conduction cooled eurocards	WS.GV.3.5.2
1D/1L	ISO/IEC 8802-3	Ethernet	
1D/1L	ICD-GPS-153	PLGR GPS ICD (Emerging Standards)	
1D/1L	EIA 485	Multi-Drop Bus	-
1D/1L	SAE AS 4075	High Speed Ring Bus	-
1D/1L	ANSI X3T10/11	Fibre Channel	C4ISR.3.2.2.2.1
1D/1L		CSMA/CD, 10/100BASE-F	-
1D/1L	IEEE 1014	Versa Module Europe (VME)	WS.GV.3.5.2
1D/1L		Universal Serial Bus (USB)	-
1D/1L	IEEE 1394 (Firewire)	IEEE standard for a high performance bus	C4ISR.3.2.2.2.2

Table E-4. (Sheet 2 of 4)

VETRONICS TECHNOLOGIES TESTBED March 2000			Joint Technical Architecture (JTA) Version 3.0 15 Nov 1999
(Draft) Interface Definition	Standard	Description	Paragraph
1D/1L	IEEE 1014-1987 "Versa Module Europe (Eurocard)" (VME)	Commercial backplane control bus for high performance systems	WS.GV.3.5.2
1D/1L	IEEE P1386-1	PCI Mezzanine card standard	WS.3.5.3
1D/1L	VITA 20-199x Draft 1.6	Conduction cooled PCI mezzanine card standard	-
1D/1L	VITA 18-199x Draft	VME pin assignments for format E form factor SEM boards and backplanes	-
1D/1L	ISO 11898	Controller Area Network (CAN) Bus	-
1D/1L		Peripheral Component Interconnect (PCI)	C4ISR.CRY.3.2.1
1D/1L		Compact PCI	WS.GV.3.5.2
1D/1L	ANSI X3T10/11	Fibre Channel	C4ISR.3.2.2.2.1
1D/1L	ISO 11898	Road Vehicle - Interchange of Digital Information - Controller Area Network (CAN) for High Speed Communication (power management)	-
1D/1L	ATR	PCB format	-
1D/1L	IEEE 1101.4a Military Format-E Form Factor Modules	PCB format EIA E700AAXA 372 pin connector	-
2D	AM-7239(E)	Vehicular Amplifier Adapter w/INC	
2D	AM-7238	RF Amplifier	
2L	VESA PnD	Display monitor plug & display	-
3L	MIL-STD-1777 (RFC 791)	Internet Protocol (IP)	2.3.2.1.1.2.1.3
3L	(RFC 768)	User Data (Datagram) Protocol (UDP)	2.3.2.1.1.2.1.2
3L	ISO/IEC 9945 IEEE 1003	Information Technology - Portable Operating System Interface (POSIX)	2.2.2.2.1.7
3L	IETF Standard 51 (PPP)	Point to Point Standards	
3L	VMF TIDP (VMF)	Data Exchange	
3L	X.500	Directory Services	2.3.2.1.1.1.2.1
3L		Simple Network Management Protocol (SNMP)	2.3.2.4.1
3L	(RFC 959)	File Transport Protocol (FTP)	2.3.2.1.1.1.3
3L	MIL-STD-1778 (RFC 793)	Transmission Control Protocol (TCP)	2.3.2.1.1.2.1.1
3L		Express Transport Protocol (XTP)	-

Table E-4. (Sheet 3 of 4)

VETRONICS TECHNOLOGIES TESTBED March 2000			Joint Technical Architecture (JTA) Version 3.0 15 Nov 1999
(Draft) Interface Definition	Standard	Description	Paragraph
3L	ITU-R BT.601	Studio encoding parameters for digital television for standard 4:3 and widescreen 16:9 aspect ratio	2.2.2.2.1.4.5.1.1
3L	ITU-R BT.1302	Interfaces for digital component video signals in 525-line and 625-line television systems operating at the 4:2:2 level of recommendation ITU-R BT.601	-
3L		Asynchronous Transfer Mode (ATM)	2.3.2.2.2.5
3L	ITU-R BT 472	Video frequency characteristics for 625-line colour or monochrome television systems	-
3L	ISO/IEC 13818 (MPEG2)	Information technology - Generic coding of moving pictures and associated audio information	2.2.2.2.1.4.5.1.1
3L	VESA XVGA	Extended VGA display monitor timing	-
3L	VESA SVGA	Super VGA display monitor timing	-
3L		X-Window System	2.2.2.2.1.2
3L	(IEEE 1295)	Motif graphical user interface development toolkit	2.2.2.2.1.2
3L/1D/1L	DY4 DMV-177	The EBC common card processor	
4D		X-Window System	2.2.2.2.1.2
4D/3D/3L/2D/2L Operating System Service		VxWorks	-
4D/3D/3L/2D/2L Operating System Service		LynxOS	-
4D/3X/3L Support System Service	WSTAWG #1	WSTAWG Operating Environment (OE) Application Programmer's Interface (API)	
4L	MIL-STD-2401 (WGS 84 Datum)	Geospatial Data Interchange	
4L	MIL-STD-2525a	Common Warfighting Symbolology	2.5.2.3
4L	WSTAWG #2	WSTAWG Weapon System Mapping Services API	
4L	FBCB2	Force XXI Battle Command Brigade and Below	
4L	WSTAWG	Position and Navigation (Pos/Nav) API	
4L		Hypertext Transfer Protocol (HTTP)	2.3.2.1.1.1.8.1
4L	ISO/IEC 14496 (MPEG4)	Coding of moving pictures & audio	-
4L	WSTAWG JVMF	Joint Variable Message Parser API	
Software Engineering Service	ISO/IEC 8652	Programming Language Ada	Not Mandated
Software Engineering Service	ISO/IEC 9899	Programming Language C	-
Software Engineering Service	ISO/IEC 14882	Programming Language C++	-

Table E-4. (Sheet 4 of 4)

VETRONICS TECHNOLOGIES TESTBED March 2000			Joint Technical Architecture (JTA) Version 3.0 15 Nov 1999
(Draft) Interface Definition	Standard	Description	Paragraph
Software Engineering Service	ISO 12207	Software Development and Documentation	
Service	MIL-STD-882	System Safety Program Requirements	
Service	MIL-STD-1472	Human Engineering Design Criteria for Military Systems, Equipment and Facilities	
Service	MIL-HDBK-759	Human Factors Engineering Design for Army Materiel	
Service	MIL-STD-1815	Reference Manual for the Ada Programming Language.	
Service	MIL-STD-1275	Characteristics of 28 Volt DC Electrical Systems in Military Vehicles	-
Service	MIL-STD-704	270V dc Aircraft Electrical Power Requirements Includes 270 volt systems	-
Service	To be defined	600 volt systems	-
Service	ISO/IEC 12207	Software lifecycle processes	-
Service	IEEE/EIA 12207	Industry Implementation of ISO/IEC 12207	-
Service	J-STD-016	Software lifecycle processes, software development	-
Service	n/a	JAVA	2.2.3.4.2 (JVM)
Service	n/a	CORBA	2.2.2.2.1.11.2
Service	JSP 101	Security	-
Service	ISO/IEC 7498-1	Information Technology - Open Systems Interconnection - Basic Reference Model: The Basic Model	-
Service	ISO/IEC 8822	Information Technology - Open Systems Interconnection - Presentation Service Definition	-
Service	ISO/IEC 10746 (various parts)	Information Technology - Basic Reference Model of Open Distributed Systems	-
Service	AS 4893	Generic Open Architecture (GOA) Framework	WS.3.1
Service	ISO/IEC 8824-2	Information Technology - Abstract Syntax Notation One (ASN.1): Information Object Specification	-
Service	ISO/IEC 8825-2	Information Technology - ASN.1 Encoding Rules: Specification of Packet Encoding Rules (PER)	-
Communication Service 3L	MIL-STD-188-220B	Combat Net Radio Networking	
Service	(IEEE 1295)	Motif graphical user interface development toolkit	2.2.2.2.1.2

Table E-5. (Sheet 1 of 2)

Data	Description	Representation
brake_cmd	Commanded percentage position for the brake actuator	32-bit float
steering_cmd	Commanded percentage position for the steering actuator	32-bit float
throttle_cmd	Commanded percentage position for the throttle actuator	32-bit float
trans_gear_cmd	Commanded position for the transmission gear	enumeration
brake_pos_cmd	Commanded position of the brake actuator	16-bit float
steering_pos_cmd	Commanded position of the steering actuator	16-bit float
throttle_pos_cmd	Commanded position of the throttle actuator	16-bit float
trans_fb_pos_cmd	Commanded position of the front/back actuator	16-bit float
trans_ss_pos_cmd	Commanded position of the side/side actuator	16-bit float
brake_status	Current percentage position of the brake actuator	32-bit float
steering_status	Current percentage position of the steering actuator	32-bit float
throttle_status	Current percentage position of the throttle actuator	32-bit float
trans_gear_status	Current position of the transmission gear	enumeration
brake_pos_status	Current position of the brake actuator	16-bit float
steering_pos_status	Current position of the steering actuator	16-bit float
throttle_pos_status	Current position of the throttle actuator	16-bit float
trans_fb_pos_status	Current position of the front/back actuator	16-bit float
trans_ss_pos_status	Current position of the side/side actuator	16-bit float
drive_abort_activated	Current drive abort state	enumeration
command_failed_id	Specifies what type of command failed	enumeration
command_failed_seq_num	Specifies which command had failed	16-bit integer
drive_abort_activated_hi_lo	Current power state of the drive abort circuit	16-bit integer
headlights_cmd	Commanded state of the headlights	enumeration
blackout_lts_cmd	Commanded state of the blackout lights	enumeration
brake_lts_cmd	Commanded state of the brake lights	enumeration
smoke_gen_cmd	Commanded state of the smoke generator	enumeration
drive_image_select	Simulated commanded state of the drive sensor	enumeration
headlights_status	Current state of the headlights	enumeration
blackout_lts_status	Current state of the blackout lights	enumeration
brake_lts_status	Current state of the brake lights	enumeration
smoke_gen_status	Current state of the smoke generator	enumeration
system_voltage	Current output of the engine generator	32-bit float
fuel_level	Current fuel level	16-bit integer
eng_oil_pressure	Current engine oil pressure	32-bit float
eng_coolant_temp	Current engine coolant temperature	32-bit float
engine_status	Current engine running state	enumeration
vehicle_speed	Current vehicle speed	32-bit float
odometer	Current odometer reading	32-bit float

Table E-5. (Sheet 2 of 2)

Data	Description	Representation
drive_image_status	Simulated state of the drive sensor	enumeration
headlights_status_hi_lo	Current power state of the headlights circuit	16-bit integer
blackout_lts_status_hi_lo	Current power state of the blackout lights circuit	16-bit integer
brake_lts_status_hi_lo	Current power state of the brake lights circuit	16-bit integer
smoke_gen_status	Current power state of the smoke generator circuit	16-bit integer
system_voltage_filtered	25% of the voltage representing the system voltage	16-bit float

Conclusion

- By using the DoD TRM guidance document, TARDEC was able to identify interface classes that enable us to conduct architectural studies for the successful development of the VTT.
- The DoD TRM assisted in establishing common terms, interfaces, and service definitions in which to address interoperability issues and commonality via this singular framework.
- The benefit from using the DoD TRM in the creation of VOSA is best represented by the achievement of two fundamental categories: software portability and systems interoperability.
- The DoD TRM reference model was a useful tool in defining the analysis process for this study and it helped improve communication between members within VETRONICS.
- Addressing real-time interfaces was facilitated by the DoD TRM and its interface class definitions.
- This same VTT/TRM approach can be adapted to other military vehicles to expand and establish greater interoperability among operator stations across a wider range of military vehicles as well as air platforms (i.e., helicopters).
- Using the DoD TRM with other aids facilitates source selection in military procurements of this type as the TRM can be used as another parameter/metric to assess a contractor's approach and understanding of the problem.

References

1. Department of Defense Technical Reference Model, Version One, November 5, 1999
2. Joint Technical Architecture, Version 2, November 15, 1999
3. Software Requirements Specification for the VETRONICS Technologies Testbed, Version 2, September 22, 1999
4. System Design Description for the VETRONICS Technologies Testbed, Draft Version 3, June 30, 1999
5. System Specification for the VETRONICS Technologies Testbed, Version 3.1, August 31, 1999

Case Study 6: USPTO Technical Reference Model Case Study

U.S. Department of Commerce Patent Trademark Office

Introduction

The United States Patent and Trademark Office (USPTO) is one of 14 agencies within the Department of Commerce (DoC). USPTO's mission is derived from the U.S. Constitution, which states, "The Congress shall have the power to promote the progress of science and the useful arts, by securing for limited times to inventors exclusive rights to their discoveries." The USPTO enterprise includes six business areas: Patents, Trademarks, Information Dissemination, Corporate Support Function, Policy Function, and Infrastructure. The primary services USPTO provides include examining and issuing patents, examining and registering trademarks, and disseminating patent and trademark information to promote an understanding of intellectual property protection and facilitate development and sharing of new technologies worldwide.

To successfully carry out its mission, USPTO is supported by an array of automated information systems (AISs) and a robust underlying information technology (IT) infrastructure. USPTO's current inventory lists approximately 44,000 commercial off-the-shelf (COTS) hardware assets and 780 COTS software products. The backbone of the current IT infrastructure is a high-speed switched network that links several hundred UNIX and Windows NT Servers, more than 8,000 Windows NT desktop workstations, 140 high-speed shared printers, and in excess of 70 terabytes of on-line magnetic mass storage.

Under the Office of the Chief Information Officer (OCIO), USPTO centralizes the responsibility for IT planning, technical direction, oversight, policy formulation, system development and acquisition, day-to-day operational management of the IT infrastructure, and dissemination of IT products and services. To provide a firm foundation for carrying out these duties, OCIO has instituted a comprehensive IT Architecture Management Framework and an enterprise Technical Reference Model (TRM).

Purpose/Objective

The remainder of this document presents a case study of USPTO's implementation of its IT Architecture Management Framework and TRM. Section 3 explains the rationale and objectives for implementation. Section 4 provides insight into USPTO's methodology and approach. Section 5 quantifies the effectiveness of the USPTO IT Management Framework and TRM in meeting objectives. Section 6 summarizes lessons learned by USPTO, as well as future plans. Sections 7 and 8 provide a list of references and acronyms, respectively.

Rationale

As it evolved over the years, USPTO's IT infrastructure came to comprise a collection of heterogeneous computing and communications hardware and often incompatible COTS software products. IT tended to be managed on a piecemeal basis, with the IT infrastructure managed as a set of distinct application systems, each with its own dedicated operations and support organization. This environment constrained USPTO's ability to readily comply with new Federal laws and regulations for IT, rapidly incorporate and leverage new technologies to meet evolving requirements, and carry out its mission in the most cost-effective manner.

Recognizing these shortcomings, USPTO began in early 1994 to aggressively address its IT infrastructure problems and to better prepare itself for the challenges of the future. USPTO developed an ambitious strategic agenda to modernize and to migrate its IT infrastructure to a standards-based, open system environment that would help position it to operate more successfully and efficiently in the 21st century. USPTO's strategic agenda is focused on providing an IT environment for itself, its international partners, and the public where patent and trademark information is created once, managed effectively, used often, and evolved over time to electronic commerce whereby most transactions are performed electronically using the Internet.

USPTO decided that an enterprise-wide management approach to the IT architecture would be the most effective for satisfying strategic business area goals and meeting day-to-day operational requirements. USPTO began to formulate and implement an IT Architecture Management Framework designed to meet the key goals shown in [Figure E-18](#). Implemented effectively, this framework will ultimately lead to reductions in patent and trademark cycle times, elimination of paper-based processing, and evolution of the businesses to electronic commerce and an electronic workplace.

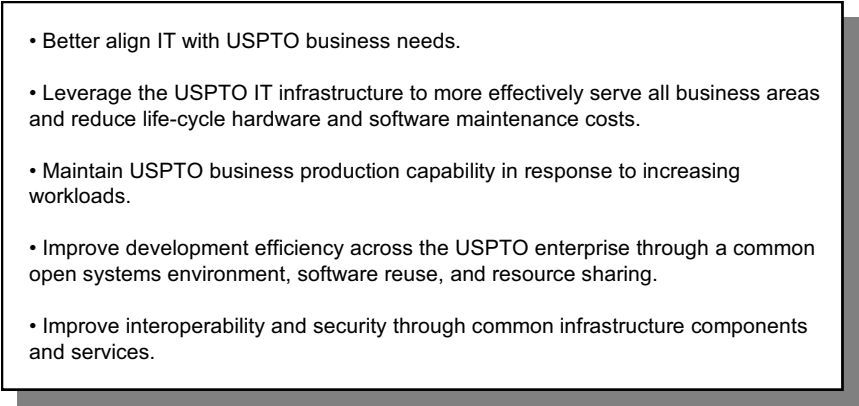
- 
- Better align IT with USPTO business needs.
 - Leverage the USPTO IT infrastructure to more effectively serve all business areas and reduce life-cycle hardware and software maintenance costs.
 - Maintain USPTO business production capability in response to increasing workloads.
 - Improve development efficiency across the USPTO enterprise through a common open systems environment, software reuse, and resource sharing.
 - Improve interoperability and security through common infrastructure components and services.

Figure E-18. Key Goals of USPTO IT Architecture Management Framework

Description

USPTO's IT Architecture Management Framework, illustrated in [Figure E-19](#), is consistent with the Federal Enterprise Architecture Framework endorsed by the Chief Information Officers Council as a road map for achieving better alignment of technology solutions with business mission needs.

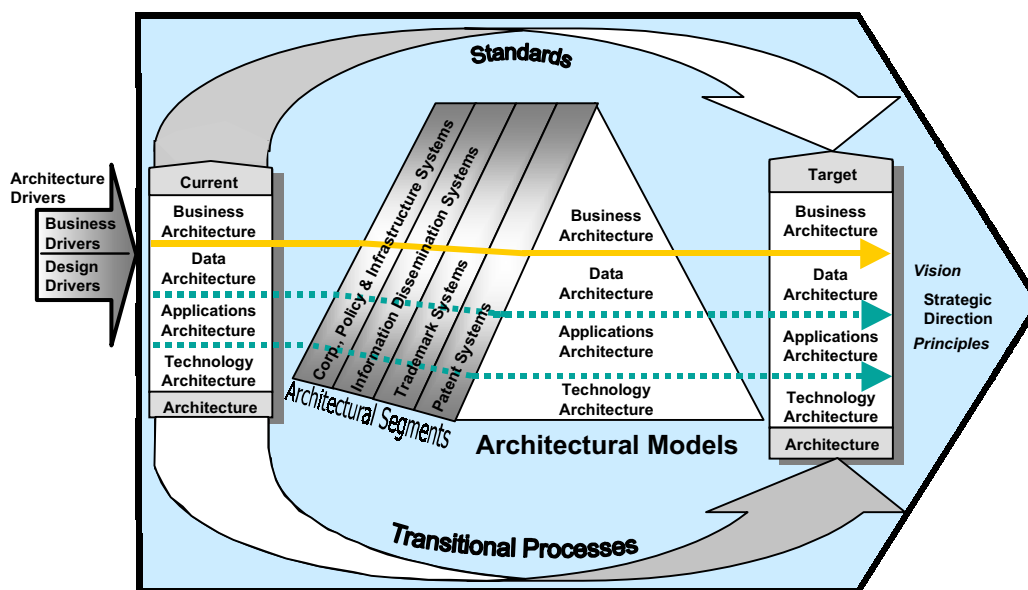


Figure E-19. USPTO IT Architecture Framework

As shown in [Figure E-19](#), the USPTO framework provides the foundation for maintaining and evolving the Current (“As-Is”) USPTO Architecture to the Target (“To-Be”) Architecture needed to satisfy USPTO’s strategic direction and business goals. The framework relates the:

- External drivers, or stimuli, that cause the architecture to change
- Business functions performed in all six business areas (segments)
- Data needed to perform the business functions
- Applications needed to capture and manipulate the data
- Infrastructure technology (hardware, networks, and communication devices) needed to run the applications
- Set of standards, voluntary guidelines, and best practices in use
- Models, plans, policies, governing strategies, and other documentation used as the basis for managing and implementing changes in the architecture
- Processes that support transition from the Current to the Target Architecture
- Strategic direction guiding development of the Target Architecture.

Additionally, the framework supports OCIO in ensuring that all USPTO IT initiatives:

- Are managed in accordance with USPTO life cycle management (LCM) principles and practices. Application of these principles and practices helps deliver quality systems that meet or exceed customer expectations, work effectively and efficiently within the current and planned IT infrastructure, and are more cost-effective to maintain and enhance.
- Are consistent with USPTO's Strategic Information Technology Plan (SITP). The SITP provides important information on USPTO's strategic vision for implementing an electronic workplace, as well as its associated management strategies, planning assumptions and constraints, decision making processes, and priorities.
- Adhere to current and planned IT standards and products contained in USPTO's TRM. The TRM provides a comprehensive set of IT services, standards, and preferred products and is used for the acquisition, development, and support of all USPTO AISs and the underlying IT infrastructure. Because it is explicitly tailored to the USPTO enterprise and contains additional elements such as the standards profile and preferred products, the USPTO TRM is a more encompassing document than the DoD TRM. The DoD TRM document is intended only to provide a description of the DoD's Technical Reference Model. Its scope is necessarily broader so as to accommodate a wider range of requirements and system configurations.

[Figure E-20](#) illustrates USPTO's IT Management process. As shown, the USPTO Corporate Performance Plan and the Strategic IT Plan drive the process. [Figure E-20](#) also emphasizes the critical role the TRM plays within USPTO's IT Architecture Management Framework. By guiding and coordinating IT infrastructure acquisitions, AIS development and deployment, operations, and support across the USPTO business areas, the TRM promotes open system design.

The structure of the USPTO TRM is shown in [Figure E-21](#). The TRM includes a broad set of service areas and interface relationships that are used to increase interoperability and compatibility across USPTO systems and promote software reuse.

The first version of the USPTO TRM was published in November 1995. It was based on open system standards identified in the Department of Defense (DoD) Technical Architecture Framework for Information Management (TAFIM) and the National Institute of Standards and Technology (NIST) Application Portability Profile (APP), as well as de facto industry standards. Further tailoring was performed to include service areas, standards, and products needed to meet USPTO-unique requirements, in particular, for Document Management and Full-Text Search.

The USPTO TRM is an evolving entity that is maintained throughout each year and republished annually. The strategic direction reflected in the TRM is formally established by the CIO and is subject to Technical Review Board (TRB) and Software Engineering Process Group (SEPG) approvals. Transition from the Current to the Target Architecture is supported by revising and adding new service areas, standards, and products to the TRM. The USPTO SITP is used to establish the priorities for TRM service areas needed to support mission-critical business system development.

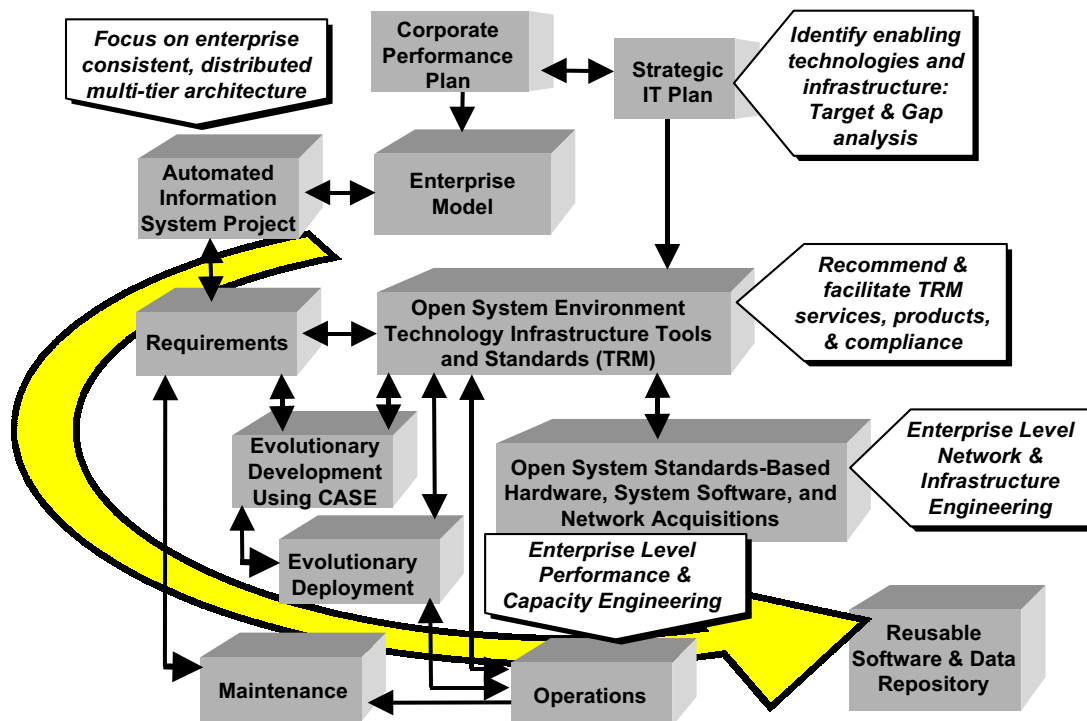


Figure E-20. USPTO IT Management Process

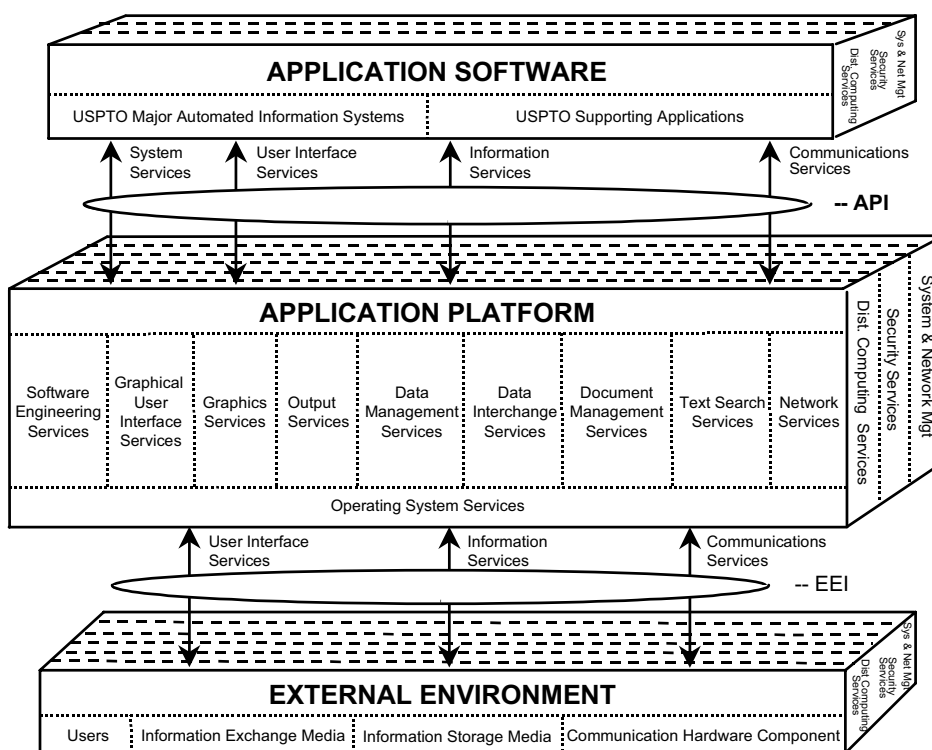


Figure E-21. USPTO TRM Service Areas

Results/Findings

Through implementation of the IT Management Framework and TRM, USPTO has made measurable progress towards the objectives shown in [Figure E-18](#). A full and complete discussion of results is provided in the USPTO publication, *Business Case: Managed Information Technology Infrastructure*. Among the important benefits realized by USPTO are:

- An improved blueprint for managing the evolution of IT in response to changing business needs
- More cost-effective IT-related expenditures due to standardization and an improved understanding of the relationship of IT to USPTO strategic and business goals
- Enhanced information flow through the USPTO enterprise and with external customers and organizations
- A proven foundation to help meet future IT challenges faced by USPTO, such as security, e-commerce, 24 x 7 operations, and on-line backup and recovery.

USPTO studies have quantified significant economic benefits. Between fiscal year (FY) 1996 and FY 2000, USPTO achieved a “one-time” savings of \$31.7 million and recurring annual savings of approximately \$32.1 million. Cost savings have been achieved due to a range of factors, including migration to a common open system environment with common infrastructure components and services, better management of the life cycle of hardware and software, improved user productivity, and reduced IT training requirements. The number of brands of desktop workstations, for example, has been reduced from 300 to less than 10, resulting in cost savings for both procurement and maintenance.

USPTO also has a significantly lower Total Cost of Ownership (TCO) for Desktop Workstations as compared to industry average. USPTO’s five-year TCO for Desktop Workstations is just under \$32,000. A November 1998 Gartner Group study found that the five-year Desktop Workstation TCO for private industry was \$44,250.

Moreover, as illustrated in [Figure E-22](#), USPTO’s improved understanding of the relationship of IT to business goals has allowed USPTO to maintain business production capabilities in response to increasing workloads. Note the significant increases in IT Workload and Search Data Base size accommodated by relatively small proportional increases in IT investment.

Between FY 1996 and FY 2000, USPTO experienced a 90 percent increase in patent image searching, 81 percent increase in patent/trademark text searching, and a three-fold increase in the size of patent/trademark search databases. Over the period FY 1996–99, USPTO also experienced a 70 percent increase in patent/trademark business transaction processing. These significant increases have been accommodated by stable IT investment expenditures.

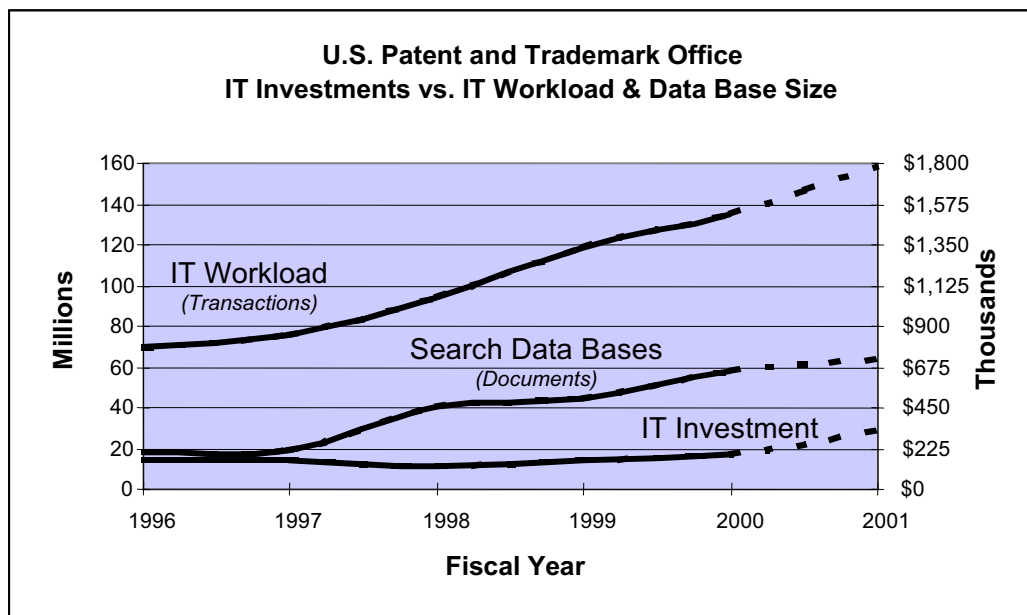


Figure E-22. USPTO IT Investments vs. IT Workload and Data Base Size

USPTO has also improved development efficiency through reuse of software components and designs, resource sharing, and the promulgation of standard data elements. According to USPTO studies, four AISs alone—Order Entry Management System (OEMS), Patent Application Location and Monitoring (PALM), Patent Cooperation Treaty Operations Workflow and Electronic Review (POWER), and Trademark Reporting and Monitoring, Advanced (TRAM++)—together used 291 standard data elements, thereby saving a total of more than \$90,000 in development costs.

Finally, the TRM has improved interoperability of USPTO systems by creating stable, directed technical environments that allow standards and processes to be defined and applied effectively and then quickly automated. For example, templates of messaging and security components were developed and subsequently used in the implementation of multiple AISs. Interoperability was achieved through business-level infrastructure components. The components represent a one-time development effort that can be reused. Subsequent systems merely interface to the components. As these components continue to be reused, redundant legacy data stores can be retired, along with the complex mapping and interfacing software that is now used to keep the data stores synchronized.

Conclusions

As the results and findings presented in Section 5 clearly demonstrate, USPTO's IT Management Framework and TRM have proven very effective in meeting objectives and providing a mechanism for IT governance. However, USPTO learned several valuable lessons during this process and is developing strategies and approaches to make the IT Management Framework and TRM even more effective.

Organizational Responsibilities

Teamwork is essential to delivering quality systems on time and within budget. USPTO's paradigm features close coordination and partnership between the business areas and OCIO. Each business area identifies and prioritizes "what needs to be done" in terms of business functions and performance goals. OCIO identifies how IT can be applied to help the business area achieve results while maintaining efficient and effective IT operations throughout USPTO. At USPTO the OCIO organization includes divisions responsible for developing High-Level Architectures; reviewing AIS technical designs; evolving the architecture for the USPTO-wide IT infrastructure; and evaluating and incorporating emerging technologies, standards, and products into the USPTO TRM.

TRM Evolution

Because of the rapid rate of technological change and advancement, the TRM will not always have standards and products defined in advance. It must evolve. In the areas of security and Web technologies, for example, innovative new strategies and products regularly emerge. In the past at USPTO, standards and products selection in these rapidly changing areas was influenced more by the particular individual system that happened to "get there first" than by an enterprise-wide assessment. This resulted in inconsistencies and selections that, in retrospect, may not have been optimal.

To address this, OCIO has established Technology Working Groups (TWGs) to focus on specific emerging and enabling technology areas of importance to USPTO (e.g., Security, Web, Middleware, and Backup and Recovery). TWG members are Subject Matter Experts (SMEs) from across the enterprise, who provide input on strategic direction and make specific TRM recommendations regarding service areas, standards, and products.

IT Architecture Information Dissemination

It is more important than ever that all employees within an enterprise gain a better understanding of the enterprise IT Architecture and the critical relationship of IT to strategic and business goals. USPTO provides training on the role of the IT Architecture for developers and managers. To provide a conveniently accessible mechanism for promulgating this information, OCIO also deployed a Web site that presents the USPTO IT Architecture in the context of the Federal Enterprise Architecture Framework.

Starting at the top level of the framework, shown in [Figure E-19](#), users can "drill down" to more detailed information on the business, data, applications, and technology infrastructure components of the USPTO IT Architecture. The Web site has been operational on the USPTO Intranet since early 2000. In late 2000, OCIO deployed an Internet version of the USPTO IT Architecture Web site that is accessible to the general public (URL: <http://www.uspto.gov/web/offices/cio/osae/sad/ita/index.html>). To make this mechanism even more effective and useful, OCIO plans to increase the breadth of IT Architecture information available via the Web and to comply with new Federal requirements (known as "Section 508") to make USPTO web pages more accessible to people with disabilities.

IT Architecture Maturity

In recognition of the importance of effectively implementing IT Architectures, DoC recently introduced a methodology for assessing the IT processes used by its agencies. This methodology, known as the Department of Commerce IT Architecture Capability Maturity Model (CMM), is an aid for conducting such assessments. The CMM is used to spotlight processes requiring improvement and provide a defined path toward that improvement. The DoC IT Architecture CMM includes six levels ranging from 0 (No IT Architecture Program) to 5 (Optimizing—Continuous Improvement of the IT Architecture Process). USPTO will continue to actively support DoC IT initiatives, such as the IT Architecture CMM, and strive to continuously improve its IT Architecture processes.

References

1. Chief Information Officers Council. Federal Enterprise Architecture Framework, Version 1.1. September 1999.
2. Chief Information Officers Council. CIO Council Web Page (URL: <http://cio.gov/index.htm>).
3. Department of Commerce. Information Technology Enterprise Architecture Affinity Group Web Page (URL: <http://www.hpcc.noaa.gov/docita/>).
4. Department of Defense. Technical Architecture Framework for Information Management (TAFIM), Version 3.0. Volume 2: Technical Reference Model. 30 April 1996. [Note: The DoD mandate for use of the TAFIM was rescinded on January 7, 2000.]
5. Department of Defense. Technical Architecture Framework for Information Management (TAFIM), Version 3.0. Volume 3: Architecture Concepts and Design Guidance. 30 April 1996. [Note: The DoD mandate for use of the TAFIM was rescinded on January 7, 2000.]
6. Department of Defense. Technical Reference Model (DoD TRM), Version 1.0, 5 November 1999. [Note: The DoD TRM supersedes TAFIM, Version 3.0, Vol. 2, 30 April 1996 (reference 4), and was released via DoD Promulgation Memoranda on 21 March 2000. The DoD TRM is 100% compatible with the TAFIM TRM and requires no transition.]
7. National Institute of Standards and Technology. Application Portability Profile (APP): The U.S. Government's Open System Environment Profile, Version 3.0. NIST Special Publication 500-230. February 1996.
8. USPTO. Business Case: Managed Information Technology Infrastructure. March 1, 2000.
9. USPTO. Life Cycle Management for Automated Information Systems (LCM-AIS). December 1997.
10. USPTO. Strategic Information Technology Plan (SITP), FY2000 to 2005. February 2000.
11. USPTO. Strategic Information Technology Plan Executive Overview, Fiscal Years 2000-2005. February 24, 2000.

12. USPTO. Technical Reference Model, Version 5.0. October 26, 1999.
13. USPTO. USPTO Information Technology Architecture Web Page
(URL: <http://www.uspto.gov/web/offices/cio/osae/sad/ita/index.html>).
14. USPTO. USPTO Managed Information Technology: Training Program Briefing. July 2000.
15. USPTO. Technical Reference Model (TRM) Strategy: Technical Review Board (TRB) Informational Briefing. August 2000.

Acronyms

AIS	Automated Information System
API	Application Program Interface
APP	Application Portability Profile
CIO	Chief Information Officer
CMM	Capability Maturity Model
COTS	Commercial-Off-the-Shelf
DoC	Department of Commerce
DoD	Department of Defense
EEI	External Environment Interface
FY	Fiscal Year
IT	Information Technology
LCM	Life Cycle Management
NIST	National Institute of Standards and Technology
OCIO	Office of the Chief Information Officer
OEMS	Order Entry Management System
PALM	Patent Application Location and Monitoring
POWER	Patent Cooperation Treaty Operations Workflow and Electronic Review
SEPG	Software Engineering Process Group
SITP	Strategic Information Technology Plan
SME	Subject Matter Expert
TAFIM	Technical Architecture Framework for Information Management

Case Study 7: Mapping of JTA Version 3.1 Core Standards to TRM Version 1.0 Services and Interfaces

The Aerospace Corporation

Introduction

According to the DoD Joint Technical Architecture, “The JTA is structured into service areas based on the DoD Technical Reference Model.” Because of the evolutionary nature of these two documents, developing a detailed mapping of the JTA standards to both service areas and interfaces as defined in the TRM can provide insight into ways to improve both documents in order to make them easier to understand and apply to DoD Information Technology systems. Also, a detailed mapping provides a reference correlation between the JTA standards and the TRM services and interfaces. A software developer required to implement a JTA-compliant system can utilize this correlation to determine what JTA standards are available for specific services. Finally, the principles used in developing the mapping form the basis of a valuable case study in the application of the TRM.

The document from which this case study is excerpted contains a mapping of the JTA core standards to the TRM services and interfaces. With some noted exceptions, all of the standards referenced in the JTA core are mapped to one or more services and/or one or more interfaces defined in the TRM, using the definitions of the services and interfaces in TRM Version 1.0. Our understanding of the definitions of the service areas and interfaces defined in the TRM has evolved as a consequence of the process of doing this mapping, but there are still some ambiguities raised and not resolved. Section 2 of this case study provides a list of the principles used in performing the mapping, and a discussion of the issues that arose in interpreting the TRM in various situations. Examples are provided.

Mapping Principles and Issues

This section describes the guidelines that evolved in the process of mapping from the JTA standards to TRM service areas and interfaces. For some standards, the authors could not determine definitively, either from the JTA write-up or from any documentation freely available about the standard, whether any services are identified in the standard, and/or whether any interfaces are specified. In these cases, the authors made a preliminary assessment based on whatever information was available.

Some of the situations encountered in mapping JTA standards to the TRM are listed below. For convenience, the situations are grouped into categories. Most of the standards fell into one of the categories described under General Cases below. The remaining subsections describe the other categories of standards, with a description of the approaches used to resolve the mapping for each.

1. General Cases

The most straightforward mapping is when a standard clearly identifies a service described in the TRM, including both the interfaces to it and its behavior. We considered this to be the nominal case. In this case, the standard was mapped to the appropriate TRM service area and also to a direct interface, generally 3D or 4D depending on the layer in which the service resides. In most cases, direct interfaces are provided by a service in one layer to user services in the layer above it. To illustrate, a 4D Interface is used by applications in Layer 4 to access services provided in Layer 3; a 3D Interface provides services to Layer 3 from services provided in Layer 2; and similarly for the other interfaces. However, in some situations, standards which mapped to services in the Application Platform Entity provided interfaces to other services in the same layer; those standards were mapped to the 3X Interface instead of the 4D Interface. Note: one corollary of this process is that a mapping to a direct interface cannot exist without a corresponding mapping to a service area.

(Ex: 2.2.2.2.1.3 ISO/ IEC 9075-3-1995 Information Technology—Database Languages—SQL mapped to Data Management Services and 4D Interface;
 2.2.2.2.1.4.5.1.1 ANSI/SMPTE 292M-1998, Television—Bit-Serial Digital Interface for High-Definition Television Systems mapped to Physical Environment Services and 3D Interface;
 2.2.2.2.1.11.2 OMG document orbos/ 98-06-01, CORBA services DCE/ CORBA Internetworking Service mapped to Distributed Computing Services and 3X Interface.)

Some of the standards mapped to two or more different services, particularly the Information Security standards and the Distributed System standards. The Information Security standards were considered to map to the Security Services, even though there may not be any actual services identified or any interfaces specified in the standard. Then, depending on the standard, there might be an additional service identified, and/or interfaces specified. Similarly, Distributed System standards were typically mapped to both Distributed Computing Services and also to whatever other service the particular standard identified.

(Ex: 2.6.2.2.2.2 IETF RFC-1510, The Kerberos Network Authentication Service mapped to Security Services, Communication Services and 4D Interface;
 2.2.3.1 ISO/IEC 9579: 1999 Information Technology—Remote Database Access for SQL mapped to Distributed Services, Data Management Services and 4D Interface.)

In general, standards that specify protocols were mapped to the Communication Services at the Application Platform Entity layer; if a standard also specifies an interface, then it was also mapped to direct interfaces, logical interfaces, or both.

(Ex: 2.3.2.1.1.1.1 IETF Standard 10/ RFC-821/ RFC-1869/ RFC-1870, Simple Mail Transfer Protocol (SMTP) Service Extensions mapped to Communications Services and 4D Interface;
 2.3.2.1.1.1.6 IETF RFC-951, Bootstrap Protocol mapped to Communication Services and 3L Interface;
 2.3.2.1.1.1.2.2 IETF RFC-777, Lightweight Directory Protocol (LDAP) mapped to Communication Services, and 4D and 3L Interfaces.)

2. Design Considerations

Some JTA standards do not identify any services defined in the TRM, nor specify any direct or logical interfaces. Rather, these standards specify processes, procedures, and/or guidelines for a person to follow in designing a system. We mapped this type of standard to what we tentatively refer to as Design Considerations. The main example of this type of standard is a user's or programmer's guide. Most of the standards that were mapped to Design Considerations were not also mapped to any TRM service areas or interfaces, but there were some exceptions: there were some multi-volume standards, where some volumes identified a service and/or interface, and another specified guidelines or procedures; also, some of the standards from JTA Section 2.6 (Security Standards) specified programmer guidelines, although they were considered to map to security services. This kind of standard was mapped to Design Considerations in addition to whatever other mapping was appropriate. Suggestions for other ways of dealing with these JTA standards are solicited.

(Ex: 2.2.2.2.1.2 M021 CDE 2.1/Motif 2.1 User's Guide mapped to Design Considerations;

2.3.2.1.1.1.1 IETF RFCs 2045-2049, Multipurpose Internet Mail Extensions (MIME) mapped to Communication Services, Design Considerations, and 4L Interface;

2.6.2.3.1.1 FIPS-PUB 140-1, Security Requirements for Cryptographic Modules mapped to Security Services and Design Considerations.)

3. Data and Data Formats

Standards that specify data formats are mapped to logical interfaces because they specify the format to be used for the exchange of information, usually between services in the same layer. In the case of human-readable formats, there may be no relationships specified in the standard between services, but they were still mapped to the 4L Interface.

(Ex: 2.3.2.1.1.1.8.2 IETF RFC-1738, Uniform Resource Locators mapped to the 4L Interface;

2.4.2.5.2.2 MIL-STD-6040, United States Message Text Format (USMTF), mapped to the 4L Interface.)

In some cases, the standard specifies a data format whose purpose is included in the description of a service in the TRM, even though the standard does not specify any behavior. In these cases, the relevant TRM service area is cited, even though there are no direct interfaces specified.

(Ex: 2.3.2.1.1.1.2.1 ITU-T X.500, The Directory—Overview of Concepts, Models, and Services—Data Communication Networks Directory mapped to Communication Services and 4L Interface;

2.6.2.3.1.1.2 MIL-STD-2045-48501, Common Security Label mapped to Security Services and 3L Interface.)

Standards that specified data content were mapped to the 4L Interface because this seemed like the closest fit, since the data format was also specified. A special case of that type of standard are standards that specify character sets. These standards were mapped only to the 4L Interface, since they specify a format for the representation of characters, even though, in the JTA, they are listed under a TRM service area such as Internationalization.

(Ex: 2.2.2.2.1.4.3 FIPS PUB 10-4, Countries, Dependencies, Areas of Special Sovereignty, and Their Principal Administrative Division mapped to the 4L Interface; 2.2.2.2.1.8 ANSI/ISO 8859-1:1987, Information Processing—8-Bit Single Byte Coded Character Sets, Part 1: Latin Alphabet No. 1 mapped to the 4L Interface.)

4. Other Cases

Standards that only specify algorithms are considered to map to logical interfaces because no services or direct interfaces were specified in the standard.

(Ex: 2.2.2.2.1.4.4 MIL-STD-188-196, Bi-Level Image Compression for the National Imagery Transmission Format Standard mapped to the 4L Interface.)

Standards that specify communication transmission formats, such as those in JTA Section 2.3.2.3, also identify Communication Infrastructure services. Since the standard specifies the interface to the communications link itself, not the data transmitted across it, it was mapped to the External Environment Entity. This type of standard also mapped to an associated 1D Interface.

(Ex: 2.3.2.3.1.1.1 MIL-STD-188-181B, Interoperability Standard for Single Access 5-kHz and 25-kHz UHF Satellite Communications Channels mapped to External Environment Entity and 1D Interface.)

The Java Virtual Machine allows applications to run on a variety of platforms without the need to rewrite or recompile the application. The standard that specifies the Java Virtual Machine was mapped to the Software Engineering Services because it describes a platform that enables Java applications to run on any computer without rewriting or recompiling; and to the 3X Interface because it contains a description of the interfaces required for compiled bytecode to invoke the JVM services.

(Ex: 2.2.3.4.2 Java Virtual Machine (JVM) mapped to the Software Engineering Services and the 3X Interface.)

Some sample mappings of the JTA to the DoD TRM, excerpted from the complete mapping document, follow:

Table E-6. (Sheet 1 of 4)

JTA Section and Service Area	Currently Mandated Standard	TRM Mapping	Mapping Justification	Notes
2.2.2.2.1.3 Data Management Services	ISO/IEC 9075: 1992, Information Technology - Database Language - SQL, as modified by FIPS PUB 127-2: 1993, Database Language for Relational DBMS (Entry Level SQL)	Application Platform Entity - Data Management Services and 4D Interface	This standard maps to the Data Management Services because it describes a database management language. This standard maps to the 4D Interface because it provides the ability to retrieve information from an RDBMS using SQL.	<ul style="list-style-type: none"> These services support the definition, storage, and retrieval of data elements from Database Management Systems (DBMSs). [JTA] Database management system services, which provide data administration, managed objects functionality, and controlled access to, and modification of, structured data...DBMS services are accessible through a programming language interface, an interactive data manipulation language interface such as SQL, or an interactive/fourth-generation language interface. [TRM] Direct Interface: transfer of information [TRM]
	ISO/IEC 9075-3:1995 Information Technology - Database Languages - SQL - Part 3: Call-Level Interface (SQL/CLI)	Application Platform Entity - Data Management Services and 4D Interface	This standard maps to the Data Management Services because it describes a database management language. This standard maps to the 4D Interface because the TRM defines API's to be 4D Interfaces.	<ul style="list-style-type: none"> The SQL/Call Level Interface (CLI) addendum to the SQL standard provides a standard CLI between database application clients and database servers. The following API is mandated for both database application clients and database servers. [JTA] Database management system services, which provide data administration, managed objects functionality, and controlled access to, and modification of, structured data...DBMS services are accessible through a programming language interface, an interactive data manipulation language interface such as SQL, or an interactive/fourth-generation language interface. [TRM] The API is defined as the interface between the application software and the application platform across which all services are provided. [TRM] Direct Interface: transfer of information [TRM]
2.2.2.1.4.1 Document Interchange	ISO 8879: 1986, Standard Generalized Markup Language (SGML), with Amendment 1, 1988	4L Interface	This standard maps to the 4L Interface because it describes the rules for applying a system of markup tags.	<ul style="list-style-type: none"> SGML is a meta-language, providing the rules for designing and applying a system of markup tags rather than the specific set of tags. [JTA] Logical Interface: supports understanding of information [TRM] Layer 4: Applications Software layer. [TRM]
	HTML 4.0 Specification, W3C Recommendation, revised 24-Apr-1998, Rec-html40-19980424.	4L Interface	This standard maps to the 4L Interface because it contains a specification for representing structural, presentational, and semantic information in combination with the contents of the document.	<ul style="list-style-type: none"> For hypertext documents intended to be interchanged via the Web or made available via organizational intranets [JTA] Logical Interface: supports understanding of information [TRM] Layer 4: Applications Software layer. [TRM]

Table E-6. (Sheet 2 of 4)

JTA Section and Service Area	Currently Mandated Standard	TRM Mapping	Mapping Justification	Notes
2.4.2.1 Activity Modeling	IEEE 1320.1-1998, IEEE Standard for Functional Modeling Language—Syntax and Semantics for IDEF0.	Application Software Entity - Engineering Support and 4L Interface	This standard describes a modeling language. Modeling is included in the Engineering Support service of the TRM. This standard does not map to any direct interfaces because its services are not used directly by Mission Applications components. This standard maps to the 4L Interface because it describes modeling language semantics and syntax for developing structured graphical representations to assist in the analysis of the system.	<ul style="list-style-type: none"> IEEE P1320.1, IDEF0 Function Modeling, is the standard that describes the IDEF0 modeling language semantics and syntax, as well as associated rules and techniques, for developing structured graphical representations of a system or enterprise. [JTA] Modeling and simulation services provide the capability to capture or set object characteristics or attributes and parameters of a system of objects, and to portray the relationships and interactions of the objects to assist in the analysis of the system. [TRM] Logical Interface: supports understanding of information [TRM] Layer 4: Applications Software layer. [TRM]
2.4.2.2 Data Modeling	DoD Manual 8320.1-M-1, DoD Data Standardization Procedures, April 1998 (which mandates the use of the DDM).	Design Considerations	This standard maps to Design Considerations because it contains guidelines and procedures related to the approval, development and maintenance of common data standards.	<ul style="list-style-type: none"> The activities addressed in this manual include the identification, development, review, approval, implementation, and maintenance of data standards. [http://www-datadmn.itsi.disa.mil/8320_1m1.html]
	FIPS PUB 184, Integration Definition for Information Modeling (IDEF1X), December 1993	Application Software Entity - Engineering Support and 4L Interface	This standard describes a modeling language. Modeling is included in the Engineering Support service of the TRM. This standard does not map to any direct interfaces because its services are not used directly by Mission Applications components. This standard maps to the 4L Interface because it describes modeling language semantics and syntax for developing a logical model of data to assist in the analysis of the system.	<ul style="list-style-type: none"> FIPS PUB 184 is the standard that describes the IDEF1X modeling language (semantics and syntax) and associated rules and techniques for developing a logical model of data. [JTA] Modeling and simulation services provide the capability to capture or set object characteristics or attributes and parameters of a system of objects, and to portray the relationships and interactions of the objects to assist in the analysis of the system. [TRM] Logical Interface: supports understanding of information [TRM] Layer 4: Applications Software layer. [TRM]

Table E-6. (Sheet 3 of 4)

JTA Section and Service Area	Currently Mandated Standard	TRM Mapping	Mapping Justification	Notes
2.4.2.3 DoD Data Model Implementation	DoD Manual 8320.1-M-1, DoD Data Standardization Procedures, April 1998	Design Considerations	This standard maps to Design Considerations because it contains guidelines and procedures related to the approval, development and maintenance of common data standards.	<ul style="list-style-type: none"> The activities addressed in this manual include the identification, development, review, approval, implementation, and maintenance of data standards. [http://www-datadmn.itsi.disa.mil/8320_1m1.html]
2.6.2.2.1 Application Software Entity Security Standards	DoD 5200.28-STD, The Department of Defense Trusted Computer System Evaluation Criteria, December 1985	Application Platform Entity - Security Services and Design Considerations	The TCSEC provides a standard for security features, and more importantly, system assurance. It covers product development and influences application software development. It maps to the Security Services because it addresses security requirements for application software from the acquisition stage through the development phase. It also maps to Design Considerations because it provides developers with criteria for a system's security features.	<ul style="list-style-type: none"> To provide a standard to manufacturers as to what security features to build... To provide DoD components with a metric with which to evaluate the degree of trust that can be placed in computer systems for the secure processing... To provide a basis for specifying security requirements in acquisition specifications. ... The trusted computer system evaluation criteria will be used directly and indirectly in the certification process. [http://www.radium.ncsc.mil/tpep/library/rainbow/5200.28-STD.pdf] The DGSA identifies the following security services that may need to be provided through implementations in information system components... Authentication service... Access control... Integrity service... Confidentiality service... Non-repudiation services... Availability service... System management services... Security labeling... Information security management services [TRM]
	NCSC-TG-021, Version 1, Trusted Database Management System Interpretation, April 1991	Application Platform Entity - Security Services, Application Software Entity - Database Utilities, and Design Considerations	This standard maps to the Security Services because it provides technical guidance in specifying and identifying system security, particularly for database management systems. It is mapped to the Database Utilities for this paragraph since it defines aspects of the services in this service area. Since it defines criteria and guidelines for implementing a secure database system, it is also mapped to Design Considerations.	<ul style="list-style-type: none"> The interpretations in this document are intended to be used in conjunction with the TCSEC itself; they apply to application-oriented software systems in general, and database management systems (DBMSs) in particular. Although the interpretations, as noted, are general enough to apply to any software system which supports sharing and needs to enforce access control (e.g., transaction processing systems, electronic mail systems), in the interest of simplicity, the discussion, and thus the terminology, will be directed toward DBMSs. [http://www.radium.ncsc.mil/tpep/library/rainbow/NCSC-TG-021.txt] The DGSA identifies the following security services that may need to be provided through implementations in information system components... Authentication service... Access control... Integrity service... Confidentiality service... Non-repudiation services... Availability service... System management services... Security labeling... Information security management services [TRM] Database utility services provide the capability to retrieve, organize, and manipulate data extracted from a database management system. [TRM]

Table E-6. (Sheet 4 of 4)

JTA Section and Service Area	Currently Mandated Standard	TRM Mapping	Mapping Justification	Notes
	FORTEZZA Application Implementers' Guide, MD4002101-1-1. 52, 5 March 1996	Application Platform Entity - Security Services and 4D Interface	This guide defines security services as defined in the TRM, so it is mapped to Security Services. This standard maps to the 4D Interface because it defines interfaces that can be used by an application to invoke the services provided by the Fortezza card.	<ul style="list-style-type: none"> This document appears to be a guide and complements the FORTEZZA Cryptologic Interface Programmers' Guide. Since the document is export-controlled, no direct quotes were taken from the document. Confidentiality service ensures that data are not made available or disclosed to unauthorized individuals or computer processes through the use of data encryption, security association, and key management. [TRM] Direct Interface: transfer of information [TRM]
	FORTEZZA Cryptologic Interface Programmers' Guide, MD4000501-1. 52, 20 October 1997.	Application Platform Entity - Security Services and 4D Interface	This standard maps to the Security Services because the Fortezza technology uses cryptography to provide a confidentiality service. This standard maps to the 4D Interface because it describes a set of C based interfaces that can be used to access the services of the Fortezza card.	<ul style="list-style-type: none"> This document defines the commands of the Fortezza Cryptologic Interface (CI) Library. The CI Library provides the software developer with an interface to the Fortezza Crypto Card (hereafter referred to as the "Card") while isolating the developer from the cryptologic details of the Card. [http://www.armadillo.huntsville.al.us/Fortezza_docs/cipg152.pdf] Confidentiality service ensures that data are not made available or disclosed to unauthorized individuals or computer processes through the use of data encryption, security association, and key management. [TRM] Direct Interface: transfer of information [TRM]
2.6.2.2.2.1 Data Management Services	NCSC-TG-021, Version 1, Trusted Database Management System Interpretation, April 1991	Application Platform Entity - Security Services and Application Platform Entity - Data Management Services and Design Considerations	This standard maps to the Security Services because it provides technical guidance in specifying and identifying system security, particularly for database management systems. It maps to the Data Management Services because this JTA paragraph defines services for database management. This standard maps to the Design Considerations because it defines criteria and guidelines for implementing a secure database system.	<ul style="list-style-type: none"> The interpretations in this document are intended to be used in conjunction with the TCSEC itself; they apply to application-oriented software systems in general, and database management systems (DBMSs) in particular. Although the interpretations, as noted, are general enough to apply to any software system which supports sharing and needs to enforce access control (e.g., transaction processing systems, electronic mail systems), in the interest of simplicity, the discussion, and thus the terminology, will be directed toward DBMSs. [http://www.radium.ncsc.mil/tpcp/library/rainbow/NCSC-TG-021.txt] The DGSa identifies the following security services that may need to be provided through implementations in information system components... Authentication service... Access control... Integrity service... Confidentiality service... Non-repudiation services... Availability service... System management services... Security labeling... Information security management services [TRM] Central to most systems is the management of data that can be defined independently of the processes that create or use it, maintained indefinitely, and shared among many processes. [TRM]

References

1. DoD Technical Reference Model, Version 1.0, November 5, 1999
2. Joint Technical Architecture, Version 3.1, March 21, 2000
3. Mapping of Joint Technical Architecture (JTA) Version 3.1 Core Standards to Technical Reference Model (TRM) Version 1.0 Services and Interfaces, Aerospace Report No. ATR-2001(3583)-1, 2 March 2001

Case Study 8: Migrating Stovepipe Systems to Integrated/Interoperable Platforms Using the Technical Reference Model and Object-Oriented Operational Architectures

Objective of Case Study

This case study demonstrates how both the Technical Reference Model (TRM) and an object-oriented operational view can be used to migrate disparate, stove-piped systems into an integrated and interoperable system using the ideas and concepts of spiral development, evolutionary acquisition and the DoD Architecture Framework ([Figure E-23](#)). The traditional approach to system development, testing, and deployment is to build standalone systems capable of supporting specific functionality within a given mission (i.e., space, air, missile, etc.). This design strategy laid the foundation for our current redundant system architecture, which supports many operational systems and uses different programming languages. Maintaining hardware and software upgrades on standalone architecture designs of this nature imposes major limitations. These inflexible, “stove-piped” systems cannot meet the growing information exchange requirements of today’s operational environment or provide a means to keep pace with evolving technology. The Combatant Commanders Integrated Command and Control System (CCIC2S) [previously known as the North American Aerospace Defense Command (NORAD)/United States Space Command (USSPACECOM) Warfighting Support System (N/UWSS)] project was initiated to resolving this problem of multiple, complex systems that retain an abundance of overlapping features and functions.

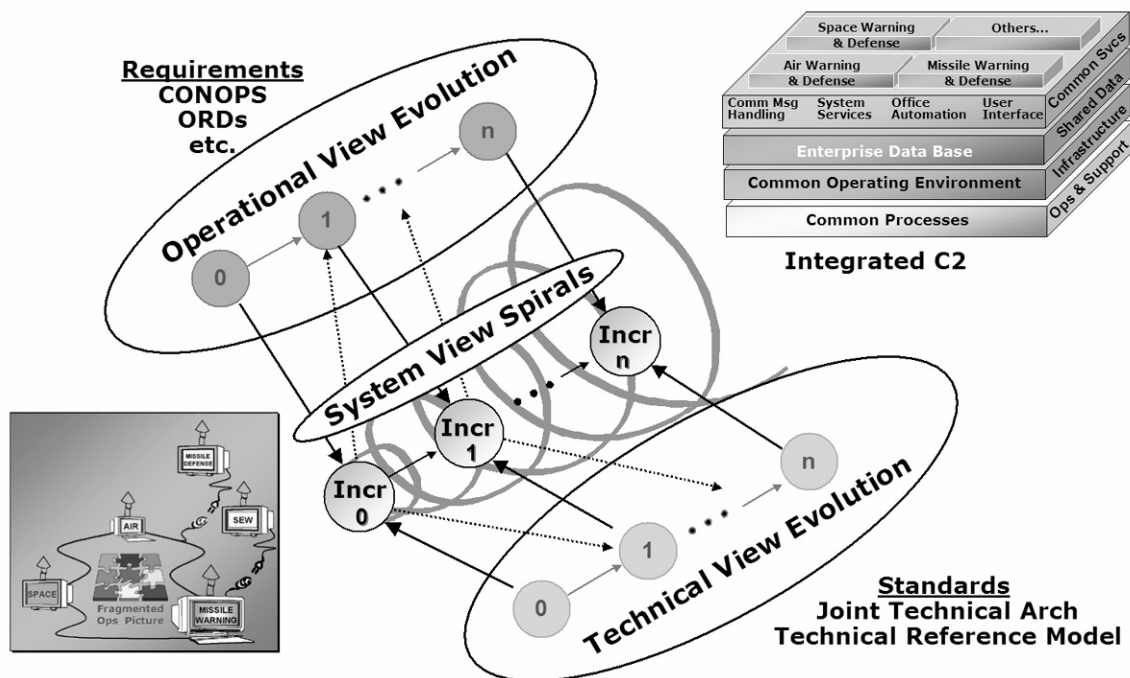


Figure E-23. Spirally Evolving to Integrated/Interoperable Command and Control

The initial stages of the CCIC2S program were focused on identifying the baseline operational environment, determining redundancies, and describing a vision for migrating current systems. The dilemma was in determining a method to examine the operational activities across mission areas while realizing or determining functional redundancies within each stove-piped process or activity. Utilizing subject matter expertise from within the Department of Defense (DoD) community and an object-oriented (OO) methodology, the CCIC2S Core Team identified and captured system redundancies and common functions within the existing Cheyenne Mountain Operations Center (CMOC) architecture, as well as the unique functions required to perform each mission.

The migration concept describes the vision and philosophy for migrating from the current complex of systems to a single multi-layered interoperable system that enables warfighters to accomplish their mission. The vision is a virtual environment that combines access to all air, space, missile, and intelligence mission information with automatic sharing of information with any authorized user who needs it worldwide.

Using the DoD TRM and a Unified Modeling Language (UML) based operational and system views, the CCIC2S Team has created an overall Department of Defense Architecture Framework (DoDAF) model that provides the migration of legacy systems to interoperability and provides the maximum level of reusability ([Figure E-24](#)).

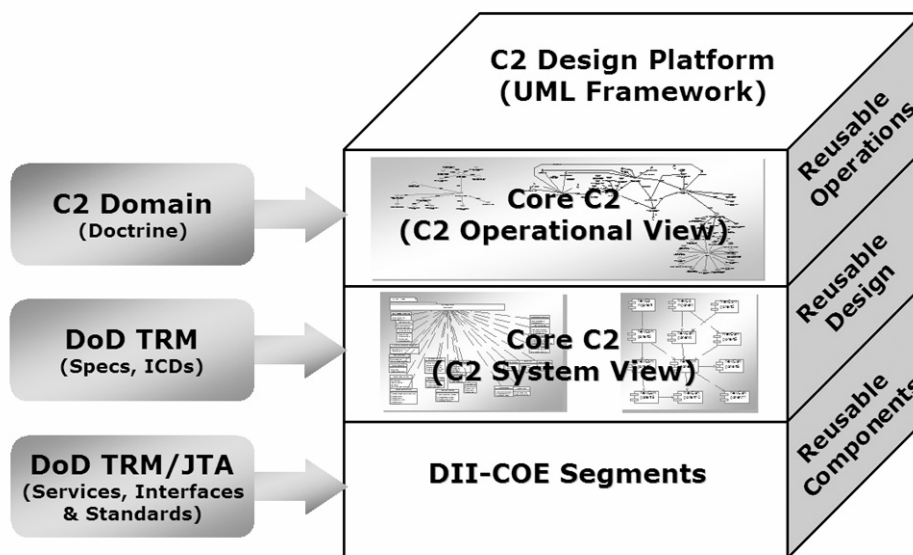


Figure E-24. Reusable Structure of the CCIC2S Architecture Approach

Operational View

The CCIC2S Operational Architecture (OA) Team used object-oriented UML to create an OA because of its robustness in symbol sets and OO characteristics such as generalization, specialization, and inheritance. UML *use cases*¹ may modify (inherit) behavior of a second *use case*; capture data interaction among operators, nodes, and systems; and allocate behavior responsibility to systems (UML objects) ([Figure E-25](#)).

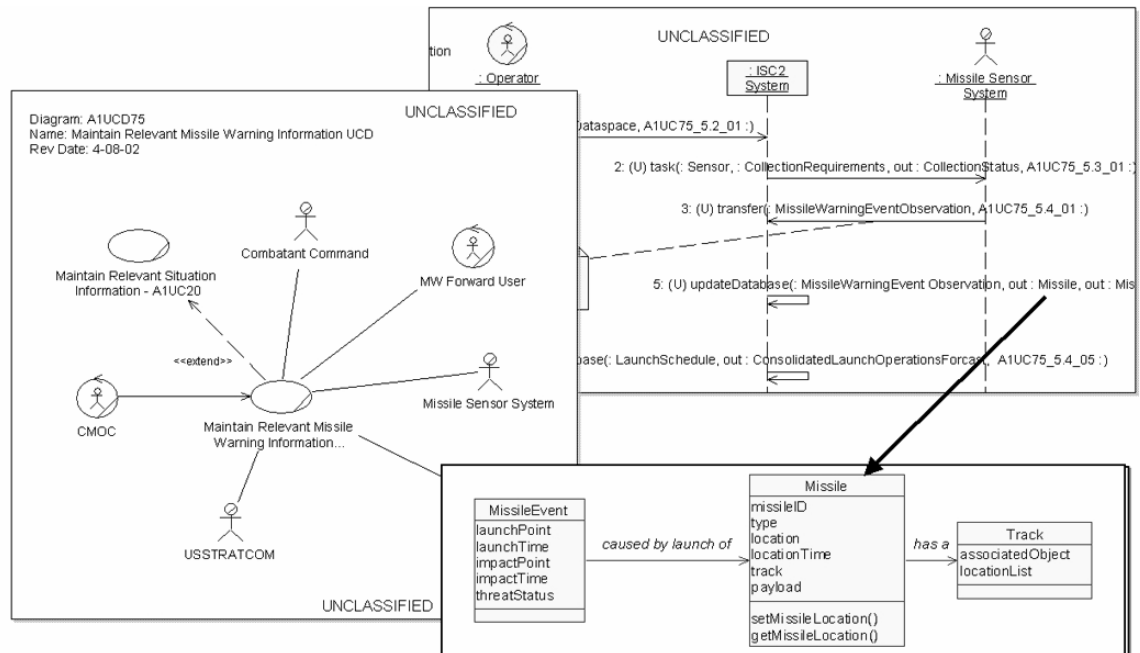


Figure E-25. Understanding and Communicating Requirements


The UML operational architecture approach provides a comprehensive understanding of operational requirements, identifies testing and training requirements earlier in system evolution, determines visually recognizable reuse, works with smaller components, provides an open design space, and focuses on system interfaces. The UML approach has a very high focus on concepts of operations (CONOPS) early in architecture evolution and the visual aspect of *use cases* provide a standardized method to evolve the system requirements. In addition, operators, subject matter experts, and stakeholders quickly grasp the UML *use case* concept. As a result, UML provides a higher level of operator understanding of operational needs by identifying *use case* observable results of value; scope; operator, element, center, organization, and system roles; and actor specification of a sequence of actions in developing the overall enterprise. Because *use cases* are highly focused on CONOPS as they evolve to requirements, testing and training planning can occur earlier in the process. Traditionally, testing and training plans are not assembled until the system reaches a maturity level near completion. Because of the nature of *use cases*, the operational analyst can understand the relationships between *use case* results of value and more easily identify design reuse in the operational process. The process also lays the foundation for developed components that are smaller and more reusable reducing the cost of potential rework. Using the OO approach to OA, the process lays the foundation for system design without imposing technological restrictions on the developer's solution. Finally, the UML process is

1. *Use case*—A description of system behavior, in terms of sequences of actions. A use case should yield an observable result of value to an actor. A use case contains all alternate flows of events related to producing the “observable result of value.” More formally, a use case defines a set of use-case instances or scenarios. The specification of a sequence of actions, including variants, that a system (or other entity) can perform, interacting with actors of the system.

highly focused on system interfaces. By focusing on system interfaces, developers can produce UML system views that show product line interaction and traditional *use case* views to design and build the system. The follow sections provide a general overview of how this business-reengineering¹ concept presents operational views useful in the development of system views.

The Operational Architecture Process

The operational architecture process begins by identifying relevant *use cases* with observable results of value (ROV) distinctiveness (e.g., [Figure E-25](#) depicts Missile Warning Information ~ ROV). Identified in the scope of the *use case*, the results of value are usually data objects (the beginning of Logical Data Model ~ OV-7 development) created or maintained by the *use case* activity. Relationships between *use cases*, if required (e.g., <<extend>>² or <<include>>³), are determined by understanding whether the extended *use case* is a modification (adaptation) of behavior⁴ of the parent case (base use case) [generally referred to as generalization/specialization⁵] or a reusable *use case* by the base *use case*. The general way to understand <<extend>> or <<include>> is that we use <<extend>> to “do on a condition of a parent case” and <<include>> “to always use a particular *use case*.” This is the foundation of use case relations in an operational level UML model and provides much payback in identifying operational patterns and reusability—essential to increase efficiency of the development activity. Ultimately, through an iterative process, the *use case* results in a System Operational Sequence (SOS) assigning behavior responsibility to the system to be built ([Figure E-30](#)) using the ideas and concepts of the Rational Unified Process for Systems Engineering (RUP SE).

Once the *use case* and its associated scope (to include ROV) are well understood, the architect, working with subject matter experts and operators, determine relevant actors and roles involved in the *use case* activity. UML actors can be organizations, centers, nodes, or systems inside [denoted by Figure E-26).

-
1. Business Reengineering—To perform business engineering where the work of change includes taking a comprehensive view of the entire existing business and think through why you do what you do. You question all existing business processes and try to find completely new ways of reconstructing them to achieve radical improvements. Other names for this are business process reengineering (BPR) and process innovation.
 2. Extend—A relationship from an extension use case to a base use case, specifying how the behavior defined for the extension use case can be inserted into the behavior defined for the base use case.
 3. Include—A relationship from a base use case to an inclusion use case, specifying how the behavior defined for the inclusion use case can be inserted into the behavior defined for the base use case.
 4. Behavior—The observable effects of an operation or event, including its results.
 5. Generalization/specialization—A taxonomic relationship between a more general element and a more specific element. The more specific element is fully consistent with the more general element and contains additional information. An instance of the more specific element may be used where the more general element is allowed.

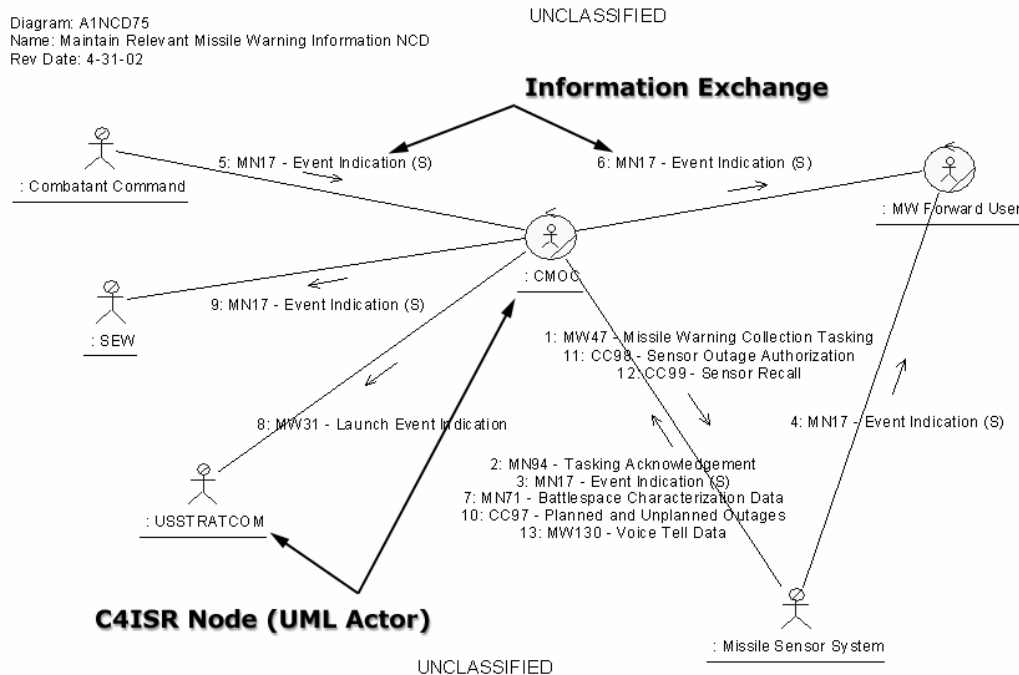


Figure E-26. Node Connectivity Description (UML Collaboration Diagram)

Based on the evolving NCD, again iteratively, the architecture team develops the NCD Sequence diagram (Figure E-27), a view of OV-5 and OV-2, that describes the desired activities, system transactions, behaviors between nodes and the underlying capabilities that depict the overall desired UML activity. Information exchanges are indicated by the UML message lines (lines with arrows) on the collaboration and sequence views (Figures E-26 and E-27). The collaboration and sequence views relate to each other at the node¹ and information exchange level.

The process continues until the *use case* relationship diagram (UCRD) represents the overall desired activities and their relationships (item 2 in Figure E-28). There are other significant views such as actor relationships (refined command relationships ~ OV-4), Information Exchange Matrix (OV-3) database, and the *use case* specification which are outside the scope of this case study discussion. Together, their views and relationships are the foundation for understanding the desired operational behavior to form the operational requirements. The overall relationships between the products views are depicted in Figures E-27 through E-33 showing product relationships 1 through 12 (shown by numbers in yellow circles).

1. Node—A representation of an element of architecture that produces, consumes or processes data.

Diagram: A1UCD75
Name: Maintain Relevant Missile Warning Information NCD Sequence
Rev Date: 4-08-02

5.1 (U) Extends Maintain Relevant Situation Information - A1UC20_E1
(U) Extends Maintain Relevant Situation Information (A1UC20) use case at the A1UC20_E1 extension point when maintaining missile warning information.

5.2 (U) Specify Baseline Information Collection
(U) CMOC analyzes the information needs associated with ongoing or planned Missile Warning operations and, using the system, specifies baseline collection for satisfying those needs. The system provides an overview of available Missile Warning data and sources (data space).
(U) CMOC selects information and sources.

5.3 (U) Task Collectors
(U) The system forwards the collection requirements to the Missile Sensor or System, receives confirmation and status of collection, and prepares to receive the collected information (e.g., allocates storage space, registers potential objects) if necessary.

5.4 (U) Update Database
(U) When collected missile warning information is provided, the system uses it to update information about missiles and missile events. The same information is simultaneously provided to the MW Forward User. USSTRATCOM requires unfiltered launch event data when it is received in significant volumes. Finally, the system provides the information to Shared Early Warning (SEW).

5.5 (U) Verify Missile Warning Output
(U) The missile warning mission must contain a verification process. It must be capable of determining that a missile warning output message is generated in response to a missile warning input and that the output message has maintained its integrity.

5.6 (U) Manage Collector Outages
(U) The system notifies the operator of planned and unplanned Missile Sensor or System outages and their estimated period. The operator may authorize the outage (planned) and recall a source to active status should priority needs arise. The sensor reports status updates when it returns to service.

5.7 (U) Return to A1UC20_5.7
(U) The use case ends.

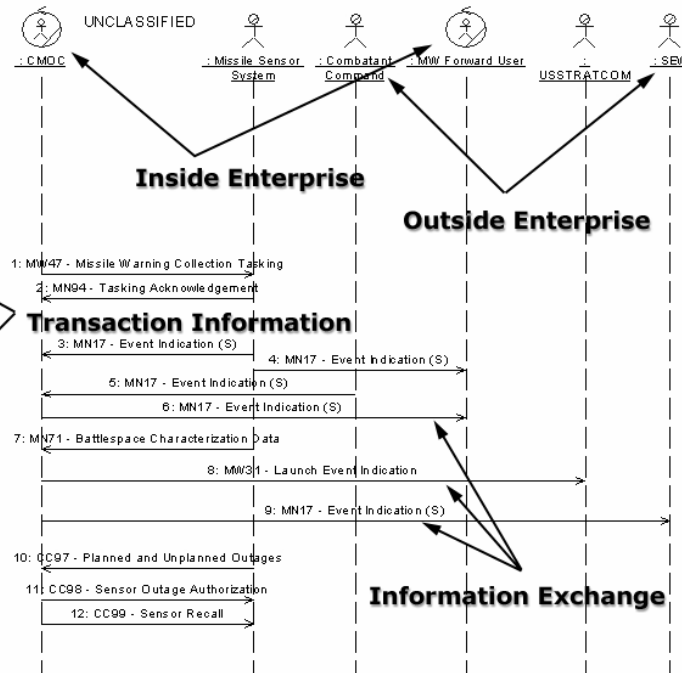


Figure E-27. NCD Sequence (UML Sequence Diagram)

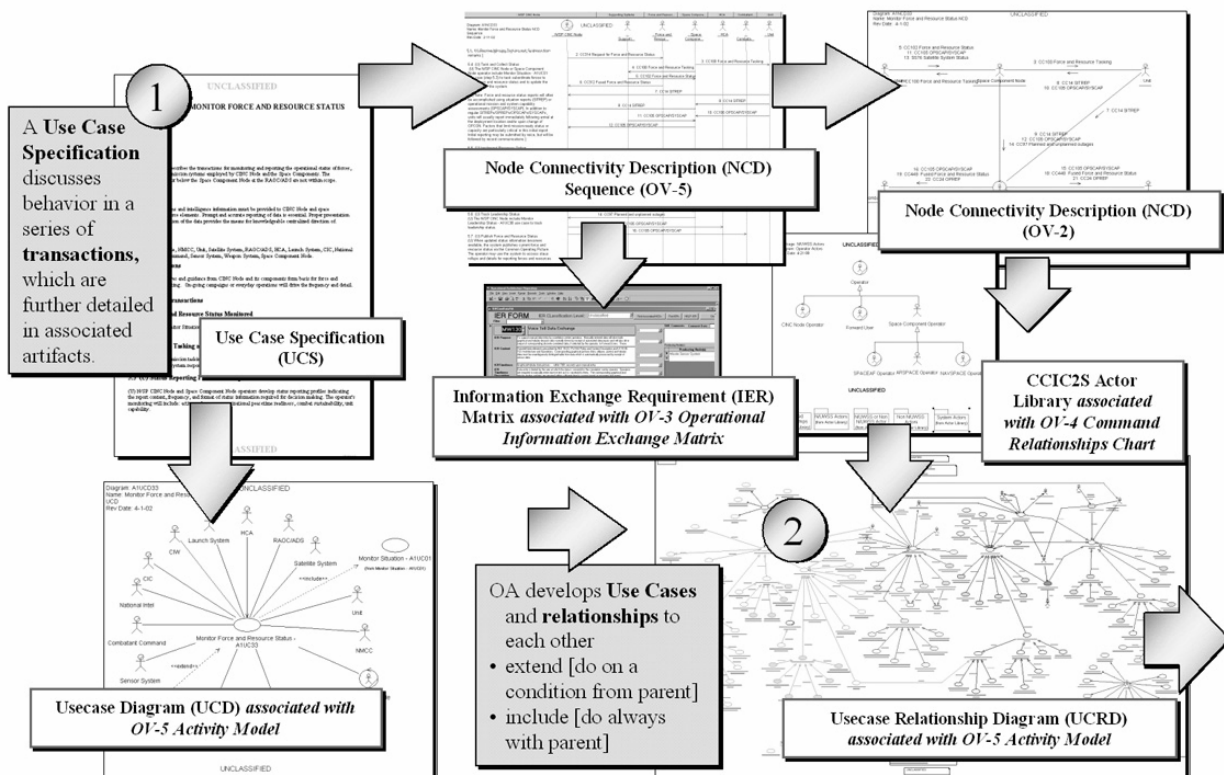


Figure E-28. Creating C4ISR Architecture Framework Operational Views

Using action verb titles derived from the primary transactions developed in the NCD Sequence (Figure E-27), the architect develops the *use case* activity diagram (OV-5) [note: one per *use case*], which show the key decision points in the operational flow and provides the view to identify consuming and producing data objects for each activity. Again, the details of this process are beyond the scope of this case study. However, because of its simplicity and visual depiction of operations, the *use case* activity diagram is a popular view with operators and stakeholders. Activity diagrams make it easier to understand the process and they show the operational flow of information necessary to support the operational activity and the underlying logical data model (~ OV-7) (Figure E-29).

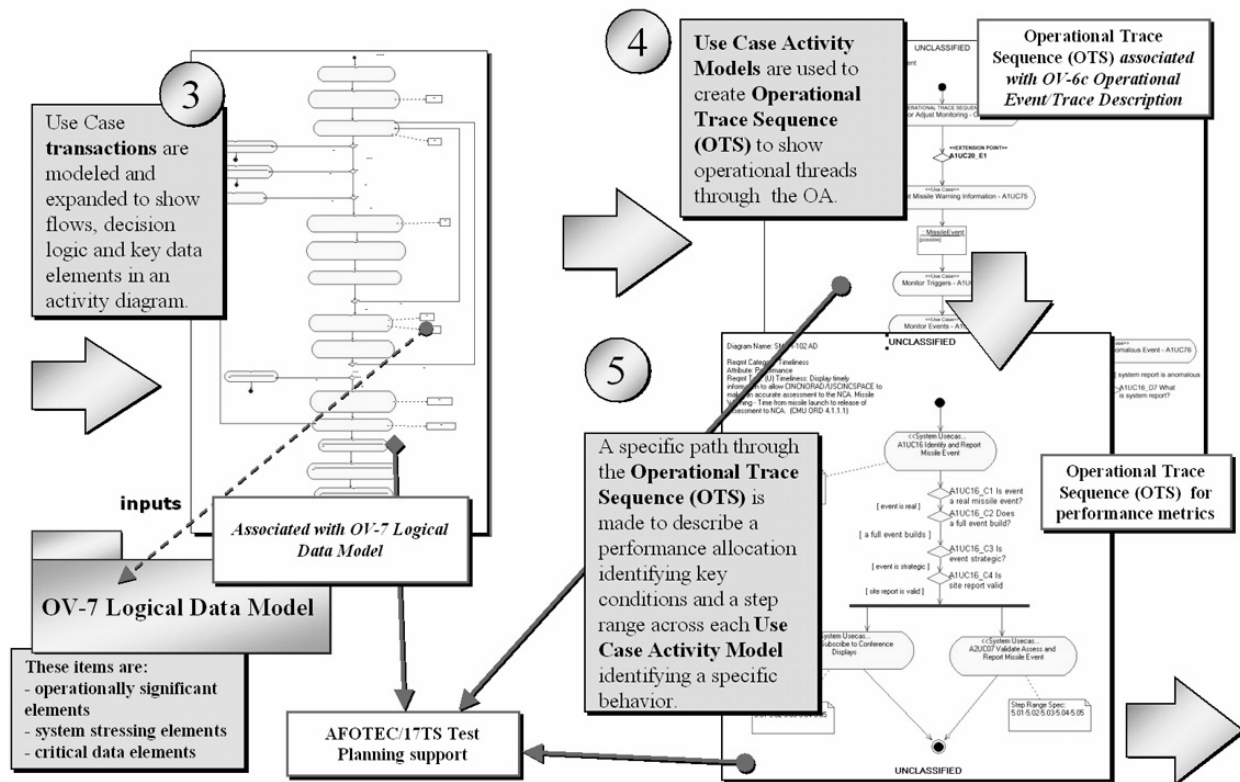


Figure E-29. Developing the Logical Data Model (OV-7) and Operational Trace Sequences (OV-6c)

Once *use cases* mature to a conceptual level to include ROV and scope, again using UML activity models, *use cases* are abstracted to a UML activity, stereotyped as <<use case>> and form the basis for building operational threads through the model. Called Operational Trace Sequences (OTS) (~ OV-6c), these high level views are useful for representing key performance parameters (KPPs), thresholds and objectives, and for building conceptual operational threads throughout the model. Operationally significant data objects and the information they contain form the information building block to describe significant operational flow through the *use cases*. Using

the OTS, we now have a way to represent CONOPS and thread the pieces of the model in a useful way. Rational Software views our OTS as a high level scenario¹ oriented *use case*. Once we demonstrated how to represent all the necessary DoD Architecture Framework views in UML, we initially lacked a way to transition from operational views to system views. As a result, we created a new view not discussed by the DoDAF called the System Operational Sequence (SOS).

Transition from Operational View to System View

To solve the transition problem, we added an additional product to those identified by the DoDAF to join the operational and system views when using object-oriented techniques. Specifically, we developed a UML sequence diagram (Figure E-30) [Note: one per *use case*] to explicitly allocate system responsibilities (transactions conveying data objects) to systems that satisfy the behavioral requirements identified in the operational view. The principles are the same as those discussed in the Rational Unified Process for Systems Engineering (RUP SE).

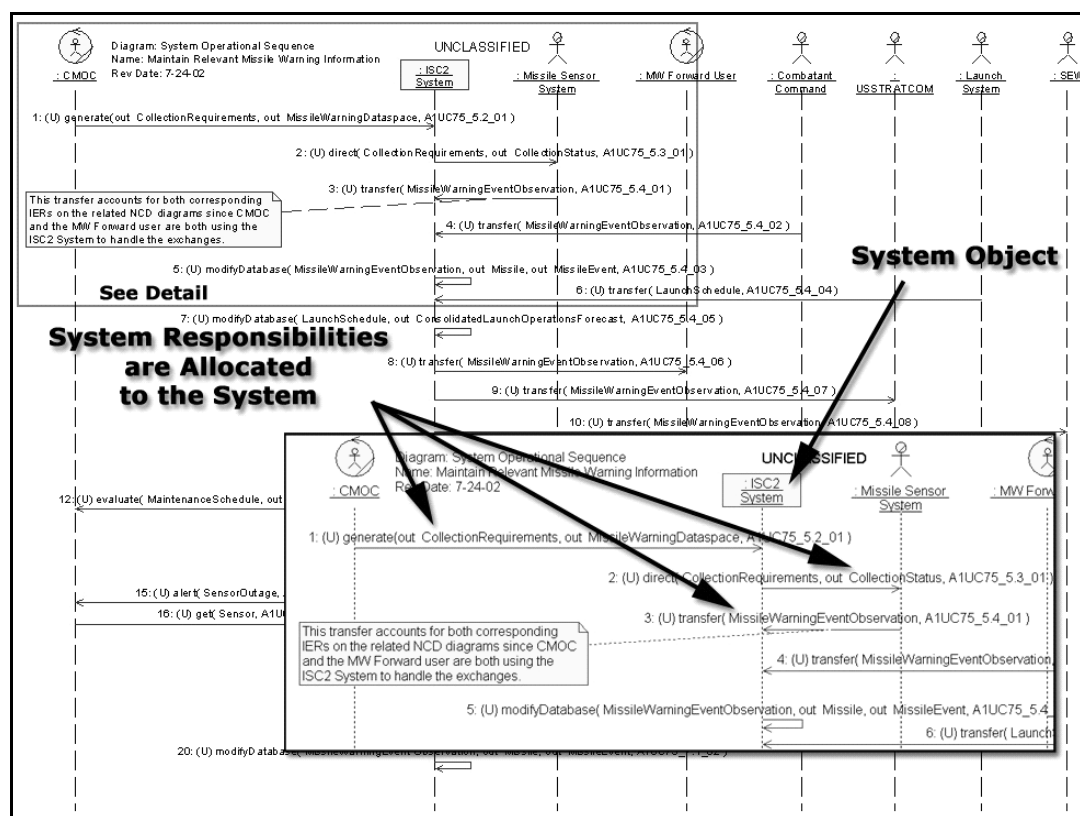


Figure E-30. System Operational Sequence

1. Scenario—A described use-case instance, a subset of a use case. A specific sequence of actions that illustrates behaviors. A scenario may be used to illustrate an interaction or the execution of a use case instance.

This product yielded several key advantages that include:

- Provides a detailed basis for tracing operational activities to system functions, contributing the operational activities identified in the SV-5 Operational Activity to System Function Traceability Matrix of the system view.
- Allows a single operational view to support multiple systems as well as the force providers and the programs that build them.
- Facilitates the expression of lower level concepts of operation.
- Clearly identifies system boundary behavior.

Thus, we can expand the scope of an operational view beyond a single system to better define cohesive operations across a whole domain or enterprise and still explicitly allocate behavioral requirements to one or more systems.

Use cases provide insight into what operational activities the system must support and to whom the supporting system capabilities must be delivered. This provides important system to node allocation information in developing the SV-1 System Interface Description and SV-2 System Communications Description.

To better facilitate requirements management and provide a way to be tool independent, we developed a Rational Rose Script called the System Responsibility Report that pulls all the information out of the SOS and builds a comma separated view (CSV) file importable into nearly any application (e.g., Excel, Access, Oracle, etc.). The added view also allows additional traceability to logical data model elements and provides a means to perform horizontal analysis on the requirements. Finally, this additional view provides additional linkage to the system view.

Linkage between the operational and system views is also established through the logical data model (OV-7). This model identifies operationally significant objects and their relationships. The system view products show inheritance or traceability to these objects via generalization or dependency mechanisms. Therefore, they directly influence the objects in the physical data model (SV-11) in a loosely coupled manner. In addition, these objects are used on the OTS to describe the object dependency in operational threads.

OV-6c, Operational Trace Sequences are used to highlight dynamics associated with key operational threads through the architecture (e.g., demonstrate how missions are supported and indicate how performance metrics apply to operations). These are associated with SV-10c, System Event Descriptions, to support testing as capabilities are fielded. The OTS describes the required behavior of the system and the System Event Description identifies what portions of the system that provides the behavior. Using these in combination, the testers can determine what behavior to test for and what system configuration to test.

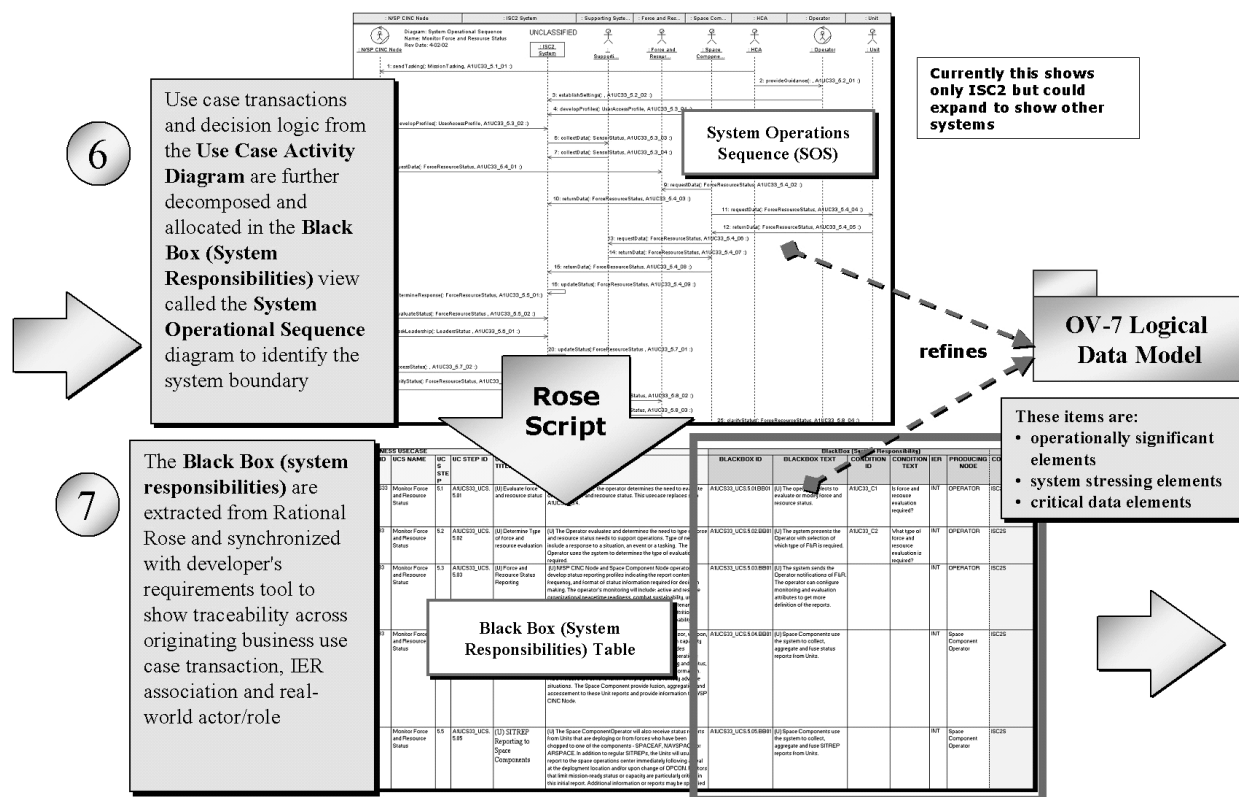


Figure E-31. Developing System Responsibilities

System View

The system view combines the elements of the technical view to provide the behavior described in the operational view. We constructed meta-models of the various products to ensure semantic linking of the operational and system views. Although our approach can accommodate multiple systems and developers, at this time we are primarily focused on the Integrated Space Command and Control System (ISC2) contractor. We adopted a layered architecture approach in concert with TRM recommendations. The system view draws on product lines and products identified in these layers to structure the components that satisfy system responsibilities allocated to the system in the operational view ([Figure E-32](#)). The traceability (from the OV generated system responsibilities (SOS ~ [Figure E-30](#)) to the system components that provide functionality) is accomplished through a recursive set of sequence diagrams, allocating the responsibilities to progressively finer grain system elements from product lines through their constituent components (see “Rational Unified Process for System Engineering [RUP SE] 1.0, a Rational Software White Paper” for discussion of such an approach).

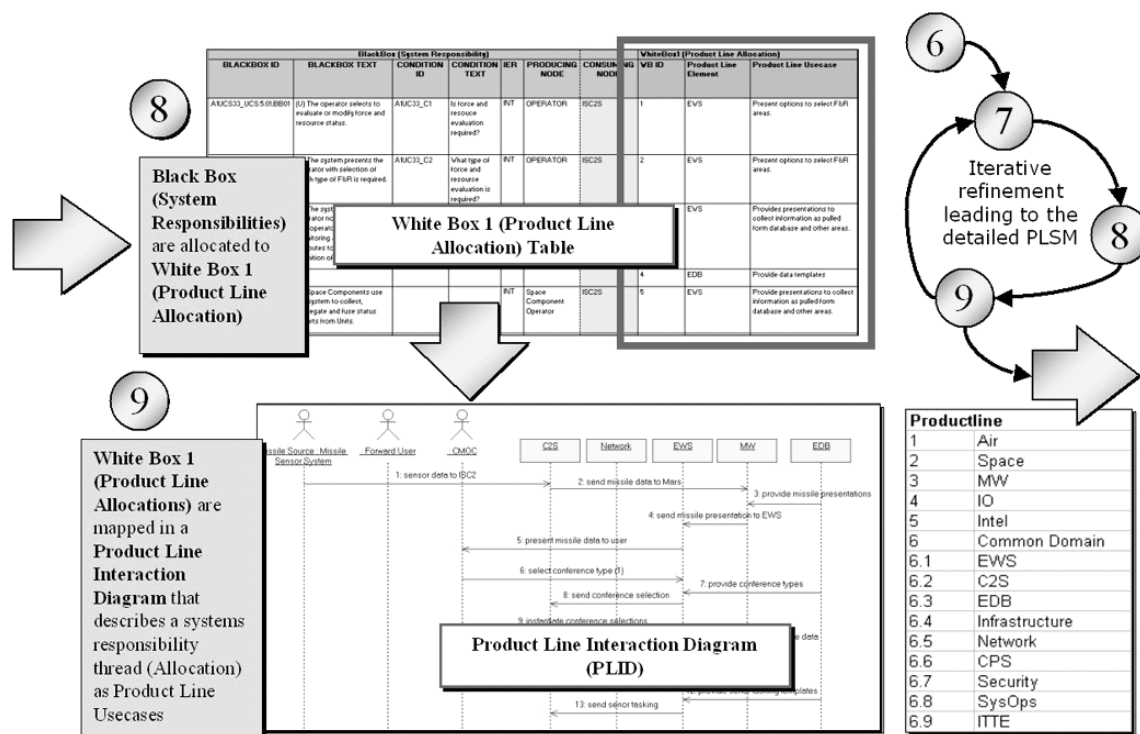
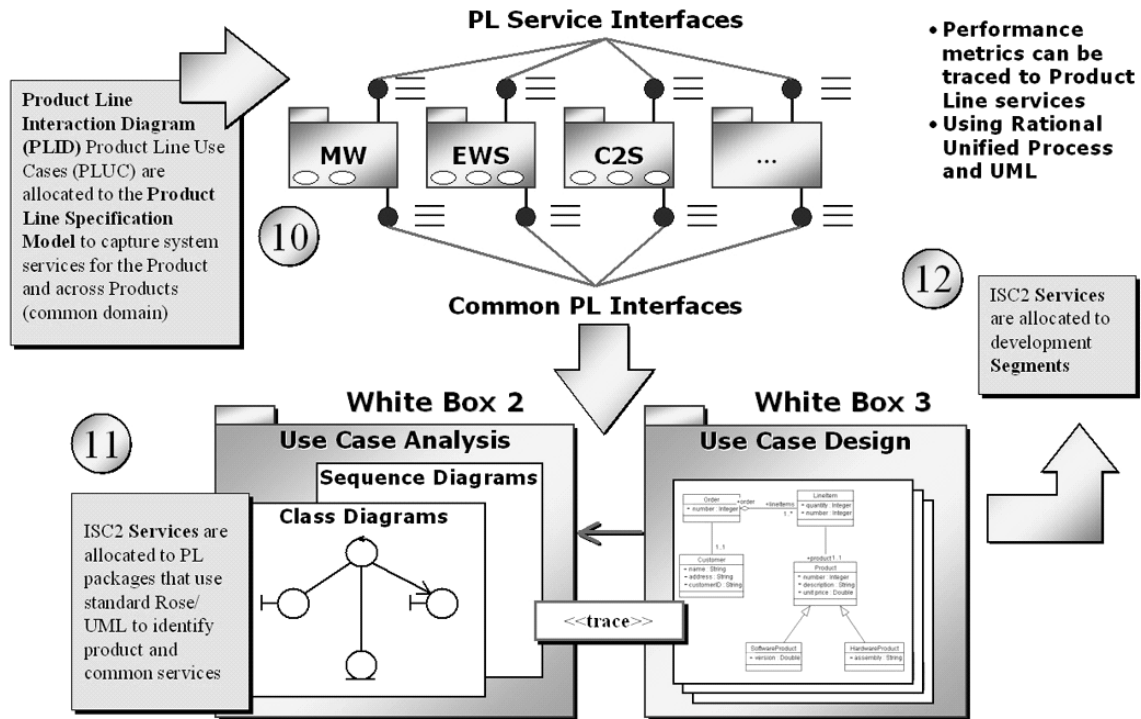


Figure E-32. Mapping System Responsibilities to Product Lines

This process establishes the architectural and design structure that ensures that the components work together to produce the required system behavior needed by operators to conduct operations. Design activities such as modeling, coding, other generation representations (e.g., XML) and roundtrip engineering then produce the code that completes the system. Using the TRM, Specifications and Interface Control Documents are applied at this level of design. Deployment views show how the software components are fielded on hardware components as well as how the later are interconnected. This becomes a further basis for SV-1, the System Interface Description, and SV-2, the System Communications Description.

While the system view describes the “to-be” visionary architecture, we are transitioning from legacy systems to future systems that embody the architecture. We must convey how the intervening mixture of legacy and future systems are deployed and cooperate to maintain continuity of operations over the development period. This is accomplished through a series of data driven deployment diagrams based on operational delivery plans.



Transition from System View to Technical View

The nominal interpretation of the Technical Architecture View is a minimal but sufficient time-phased list of standard technology specifications applicable to the realization of the system's requirements. Time phasing consists of delineating the current specifications, forecasting the major technology standardization trends and updating the associations between standards and system architecture elements as the system matures through its evolution. Since the standards are selected after related system capabilities are identified, it is natural to think in terms of a sequential process in which the specifications are selected after the system architecture has been defined. However, the real process, whether formal or not, involves intense analytical interaction between the system architecture and technical architecture domains. From the system architecture perspective, the design of both the logical and physical aspects is informed and constrained by what the architects consider practical in terms of available technology, standards and architectural patterns. From the perspective of the technical architecture, the selection of specifications from the vast domain of technology specifications must, in turn, be filtered by the context of the system architecture.

The success of this highly iterative interaction between views is currently quite dependent on the artfulness and experience of the architects. This is evident in light of orthogonal integration of emerging C2 standards, such as COE, with other major system architecture views. The future hope is that the framework will mature to provide readily accessible architecture patterns and technology specifications pre-organized by system domain.

The range of specifications and patterns that must be considered in this process within the domain of strategic command and control systems are currently dominated by component container middle-ware technology, tiered client-server patterns, object-relational mapped persistence, public key infrastructure, and communication technologies that cross the full spectrum of geographic distribution. In addition, technologies developed by the DISA Network-Centric Enterprise Services (NCES, formerly the DII COE) effort have recently become available for this domain. Significant examples of COE standards include workstation and user interface facilities, various types of message processing, and software build configuration management mechanisms. Further, each of these standard areas is experiencing strong change trends that must be forecast against the expected evolutionary life of the system.

Traceability from System View to Technical View

The problem of associating the domain's standards, patterns and their trends with logical and physical elements in the system architecture is a relational challenge for both system and technical architectures. The ISC2 developer addressed the problem with a few principles.

The first principle is called “greatest scope of technical constraint” in which a technical constraint should be mapped to the applicable system architecture element with the largest technical scope. The primary benefit of using the greatest scope principle is that the scope hierarchy inherent in the system architecture is leveraged to eliminate tedious, redundant, error-prone and probably unsustainable mappings between details of a standard and the recursive decomposition of the relevant system architecture element. (For example, user interface standards should be mapped to the enterprise workstation rather than each individual user interface display, or a security guard pattern should be mapped to the entire communication processing system rather than each individual guard element).

The second principle is “no orphan standards.” This principle is easy to understand but may be difficult to implement given the sheer size of strategic C2 domain. It can be a substantial effort to review the mapping to ensure that every standard listed in the technical architecture is indeed associated with and appropriately constrains some system architecture element. There are several obvious benefits: reduced architect and implementer learning load and reduced workload for quality assurance. A more subtle benefit is that building and cross checking the mapping for orphans provides an important cognitive review of the architecture.

These principles lead to a simple relational expression for the mapping as follows:

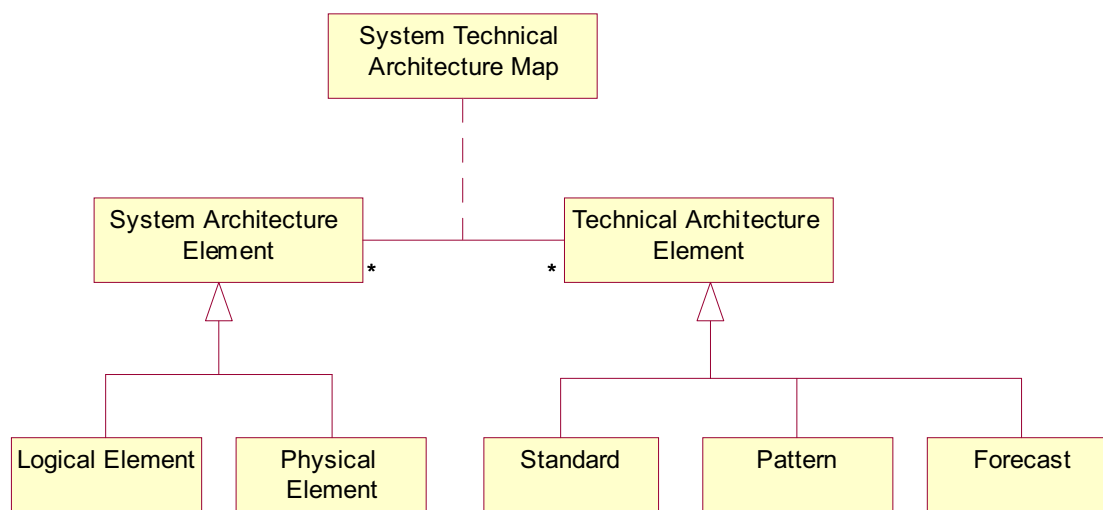


Figure E-34. Technical Architecture Map

This pattern is easily implemented using any relational tool such as a relational database (i.e., Access, Oracle, etc.).

Technical View

In summary in terms of traceability, the operational architecture provides the source of functional requirements for the CCIC2S enterprise system. The functionality identified is based on Tier 1 C2 Battle Management and support mission functions based on traditional (existing) and non-traditional (new, emerging) threats. The CCIC2S-ISC2 requirements flow-down process allows the capability to be defined, refined, aligned and allocated in terms of functional and performance requirements to selected logical systems of interest (associated with current existing or future systems). The functionality is subject to factoring, aggregation, consolidation and realignment to core system capabilities. The allocations also include the complex cross mapping with legacy (existing) systems both those that are and are not migrating and the new/emerging systems of interest. This is to focus uniquely on mission applications while enabling the reduction/consolidation of infrastructure and common/shared functionality. The next flow down association is to the technical architecture.

The JTA provides the DoD with the fundamental building codes for the Warfighter to develop the capability for interoperability, seamless information flow and plug-and-play. The CCIC2S program assimilated the JTA, the current as is CCIC2S environment and developed a minimum set of standards, the CCIC2S TA (CSTA), which is applied to the ISC2 program. This set of mandated standards and guidelines provided the starting point for the evolution of the CCIC2S enterprise systems architecture. The ISC2 developer took this set of standards and applied in-depth industry evaluations, trade studies and comparative analysis with other standards needed to achieve the defined system functionality that was allocated to ISC2 system from the CCIC2S Operational Architecture. This evaluation continues, to keep the comparative analysis current. It provides a constant forward-looking perspective to exploit new standards and technology, a constant examination of others standards within the CCIC2S enterprise to consider in the ISC2 System Architecture, and a constant scrutiny of the need for a corresponding service within the ISC2 enterprise system. The technology forecast tracks near and long-term technology trends in order to identify promising new technologies that can be effectively applied to reduce ISC2 evolution risks and costs and increase capability. New technologies are constantly evaluated from a cost/benefit standpoint to determine applicability to future ISC2 releases to deliver the most capability for the minimum cost.

The ISC2 program falls within the NCES COE environment and also the Tier 1 and lower echelon command and control (C2) business area. Currently, the Global Command and Control System (GCCS) is emerging as the core Warfighter distributed, federated C4I system. By default, the set of capabilities with the DISA-provided COE become the starting point for capabilities mapping between the associated system-define capabilities and the DISA-provided COE capabilities. The ISC2 developer is tasked to reuse, expand, enhance, tailor or build new COE capabilities following the standard DISA processes. The ISC2 developer is exploring enhancing these processes to support faster cycles for spiral evolution, for research and development and to meet user needs for short/no-notice response mission demands. The ISC2 program open standards approach emphasizes architecture constraints and driving requirements in the selected standards and technologies, which are defined in the CSTA.

The demands of legacy system migration provide a challenge to the TV-1 and the TV-2 Technology Forecast. The ISC2 developer must manage a diverse set of standards that at times conflict or cannot be applied until a certain point in the phased migration of legacy systems. And this must be done with no impact to on-going mission operations. This complex cross phasing has been described as changing an engine while in flight. The ISC2 developer has established a full cross-matrixed ISC2 product line with capability deliveries and synchronization points that align selected systems migrations/deployments. This is managed in the ISC2 Master Integrated Evolution Plan (IMEP). This evolution involves applying the CSTA to enable a core systems infrastructure and a core database infrastructure. These areas are further permuted by other CCIC2S enterprise systems (outside the current scope of the ISC2 program) that are themselves migrating/evolving and which are within the JTA but not completely compliant with the ISC2 TA. The ISC2 developer must maintain situational awareness of all external ISC2 interfaces (functionally derived from the OA) to apply a standard industry or custom technological solution to bridge these systems. The ISC2 developer uses the TV-1 and TV-2 to inform, collaborate or guide other programs that need to interface or integrate with ISC2 core systems. This common foundation also supports systems-of-systems testing, joint testing, scalable product line, flexibility in evolution and higher fidelity in a capability-component based architecture.

Additional challenges are emerging with the DISA-provided COE and the GCCS environment in terms of standards that define specific capabilities of legacy systems that are either not needed or used by the GCCS community as a whole (due to mission uniqueness) or require state-of-the-art abilities to support real-time information exchanges or capabilities (such as for battle management execution and runtime). The ISC2 developer is also working to evolve multiple environments across various missions to a common framework while being constrained by current legacy warfighter processes and warfighting paradigms. Another challenge to a common TA application is the distributed nature of the CCIC2S environment into Warfighter environments (Combatant Command/Theaters). The ISC2 developer is focusing on instantiating mission portals, either as client/server or very normalized COTS/GOTS structures to be able to respond to Warfighter mission needs, in some instances regardless of whether the Warfighter is COE compliant or non-compliant (such as web and client/server technologies). The ISC2 TV-1 and TV-2 documents have to be dynamic “living” documents and identify elements that are sustainable and affordable.

The ISC2 developer is evolving a DISA-provided COE compliant system with standard segment taxonomy structure aligned with the CSTA. The ISC2 Product Line was also designed from its inception to have a similar taxonomy of functionality partitioning aligned with the DISA-provided COE architecture including the concepts of API layers, kernel capabilities, SHADE, COTS and GOTS, Style Guides and segmentation design. The ISC2 developer is evolving the ISC2 system to meet COE compliance in both structures, constructs and processes. The development of COE applications and components each offers various options on sustainment and processes to follow. The ISC2 Developer is evolving these options successfully to facilitate spiral evolution and enhancement of migration.

Traceability from Technical View to Technical Reference Model

The purpose of the Technical Reference Model (TRM) is to provide a common conceptual framework in a defined common vocabulary of the various components of the target system. The TRM provides the taxonomy for identifying a discrete set of conceptual layers, entities, interfaces and diagrams that provides for the specification of standards. The IMEP, Section 2 – Target System Architecture is designed to be aligned with TRM constructs. The ISC2 architecture model describes the application layers, data services, distributed operations management services, middleware services, network, platform, security, and web services. The ISC2 TSA was designed to support the DoD TRM and to provide a common vocabulary to define the ISC2 open systems services and capabilities to enable interoperability, scalability, software reuse and facilitate product line manageability. The ISC2 Product Line aligns with the CCIC2S operational architecture, COE, and TRM constructs. The ISC2 architecture solution is a standards-based implementation of an E-Business system design, leveraging mature commercial capabilities to bring robust mission capabilities to any authorized Warfighter, anywhere, at any time. It also provides a high degree of flexibility and scalability to accommodate changes in CONOPS, threats, and the resultant impacts on sensors, internal and external interfaces, mission capabilities, and users. The ISC2 net-centric model shown in [Figure E-35](#) demonstrates how the ISC2 Product Line is aligned with the COE segmentation approach.

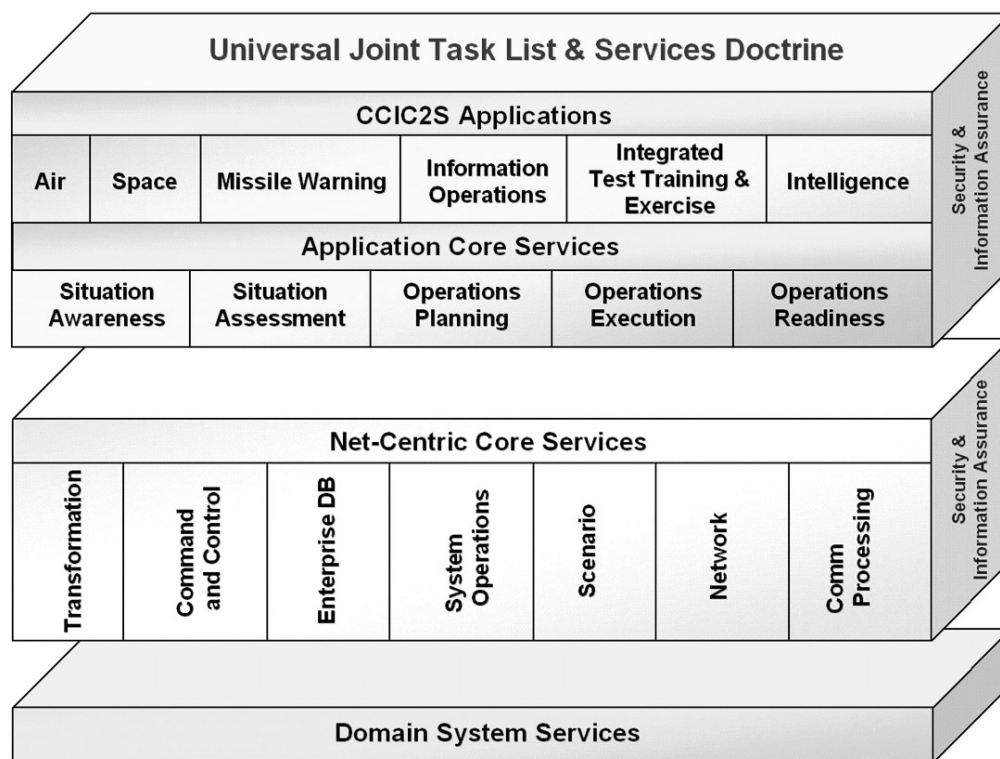


Figure E-35. Integrated Space Command and Control Net-Centric Model

The Enterprise Database is based on a single logical data model (LDM) traced from the CCIC2S Operational Architecture. These data elements are standardized according to the DoD Data Element Standardization requirements (DoD Std 8320-1), where appropriate. Existing C2 Core Reference Sets/Models are used where those definitions and ISC2 requirements coincide. The Enterprise Object Model (EOM) defines the hierarchy of the data objects and available methods to implement the C2 business rules within the ISC2 system. The example below ([Figure E-36](#)) from the IMEP, TSA section, demonstrates the Data Access Interface layer relationship between the ISC2 Logical Data Model and the Enterprise Object Model.

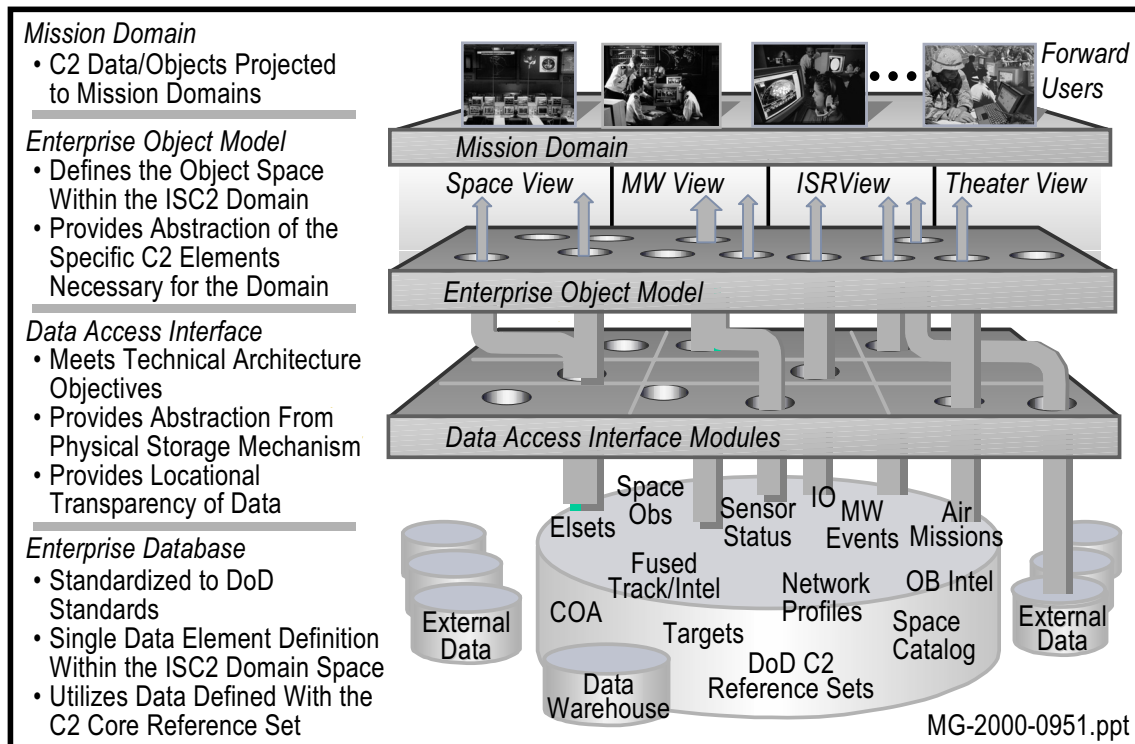


Figure E-36. ISC2 Logical Data Model and the Enterprise Object Model

The IMEP, Section 4 – Evolution visually provides target representations of the multiple system evolution by fiscal year. The system evolution is based on the physical environment including localities, devices, systems, communications infrastructures and interfaces both internal and external. The IMEP, TSA section also contains visual representations of TRM relationships across other common services, infrastructure elements and mission applications.

Conclusion

The Integrated Space Command and Control net-centric model [details depicted in the ISC2 documentation] represents the initial derivation of a domain C2 reference model using the DoD TRM. This model is in the process of being fully transitioned into the DoD Net-Centric Reference Model (NCRM) under development by the DoD TRM Working Group. Current service definitions of the ISC2 model are consistent with those offered by the NCRM and are easily accommodated within that model. As the DoD NCRM is evolved and baselined, it is expected to

enhance the ISC2 model. This parallels previous efforts in the development of the DoD TRM that subsequently resulted in the establishment of a singular referential platform-centric TRM that is tailorable for all DoD domains. The expectation in this early stage of model maturation is that identification of and convergence to a DoD Net-Centric Reference Model will facilitate the development of Net-Centric Enterprise Services (NCES) segments and other reusable software

Our team has developed a method to achieve full DoD Architecture Framework traceability while migrating to interoperable systems using the ideas and concepts of the DoD Technical Reference Model and object-oriented UML operational and system views. We have overcome many hurdles to include end-to-end traceability and the difficult migration problems to spirally evolve stove-piped systems to an interoperable common operating picture. Using industry best practice and expertise from many leading edge companies, we have teamed to solve a very complex and difficult problem that continues to agitate developers throughout the DoD (i.e., ability to trace and link requirements across the architectural views—operational, technical and system; and integrating the DoD TRM and its methodology to support interoperability and technology insertion/transition issues). Our approach, too good to be true by many, is a seamless and systematic approach to the complex problems the DoD must face to enter the net-centric solutions in the future. According to the Clinger-Cohen Act, we must look to industry for solutions. Clinger-Cohen's vision is proving itself to be a powerful way ahead for the Department of Defense.

Acknowledgment

We would like to thank the following individuals who have contributed to this case study—they include: Rob Byrd (SI International), Tom Folk (MITRE), Dr. Paul Bailor (Lockheed Martin Mission Systems), Albert Robredo (Ciber), Mike Canaday (Master Solutions). We would also like to extend a special thanks to the members of the DoD Technical Reference Model Working Group, especially William Wong (DISA), Mike Carrio (ARTEL), and Bob Starek (Navy/METOC).

References

1. C4ISR Architecture Framework, Version 2.0, 18 December 1997
2. DoD Architecture Framework Version 1.0 (draft)
3. Rational Unified Process for Systems Engineering (RUP SE) 1.0
4. Rational Unified Process, Rational Software
5. RUP SE White Paper, Rational Software
6. Information Technology Management Reform Act of 1996
7. Technical Reference Model Version 2.0, 9 April 2001

Case Study 9: LPD 17 San Antonio Class

SPAWAR Systems Center Charleston (SSCC)

Outline

- Discussion
- Objective
- Background
- C4ISR
- Interoperability
- TRM Approach
- Mapping
- Summary

Discussion

The LPD 17 USS San Antonio class ship ([Figure E-37](#)) is one of twelve ships under construction, the first of which will begin sea trials in 2004. The LPD 17 class of ships will serve as one of the chief cornerstones for US forward presence and force projection well into the 21st century. LPD 17 will serve as a task component of Naval Expeditionary Forces, and will be part of joint war fighting efforts with other services in the littoral area. The LPD 17's role will be complex from an interoperability view due to its complement of equipment, systems, and various missions. It must operate in a Common Operating Environment (COE) and Collaborative Information Environment (CIE) which will demand a robust, precise and rapid information system for command and control. Since Network Centric warfare and information superiority are key to future naval operations, a baseline architecture design must include interoperability within own units, as well as joint and multi-national forces.

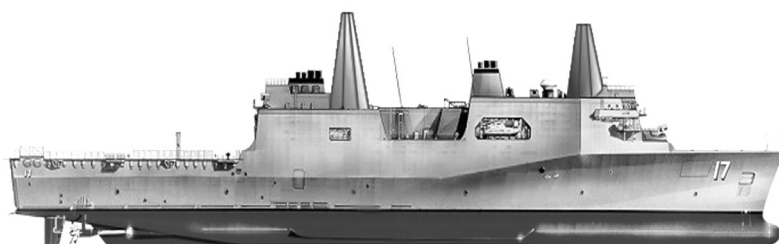


Figure E-37.

The lengthy shipbuilding schedule when compared to the shortened technology transition time period creates a technology conflict (i.e., ensuring that the latest and most mature C4ISR technology is implemented onto each LPD ship). Three major technology and interoperability issues surround the LPD 17 program:

1. Maintaining technology and interoperability synchronization with the other ships in the Amphibious Readiness Group (ARG) or Carrier Battle Group (CVBG);
2. Maintaining harmonization within the family of LPD 17 class of ships that are developed over the decade; and
3. Maintaining control and technology harmony across each LPD's development period. Developing an LPD 17 technology transition and insertion plan becomes an item of major concern to management.

Developing a risk mitigation approach that provides the visibility into and control of shipboard C4ISR equipment becomes a key instrument for ensuring interoperability and supporting missions. How to select and ensure that the best-of-breed technology and systems are integrated into the LPD 17, and at the same time synchronized with the rest of the LPD 17 class of ships under construction are major issues that will have to be addressed if interoperability is to be accomplished successfully.

Objective of Case Study

SPAWAR Systems Center Charleston (SSC-C) is charged with ensuring the operability of the engineering, design, and integration aspects of the LPD 17's C4ISR operational architecture. LPD 17's C4ISR component interoperability is crucial between the ship and the ship-wide area network (SWAN) with other Naval Expeditionary Forces, as well as other services in a joint operations environment. A Phase I methodology was developed to aid the LPD 17 C4ISR Integration Program Manager to view the relationships of all major systems, subsystems, and their components. Because LPD 17 is in a Turnkey integration shipbuilding environment, the methodology chosen will be able to provide on-going insight into COTS/GOTS products for technology insertion and facilitate adherence to standards for those products installed.

A review of the DoD Technical Reference Model (TRM) and Joint Technical Architecture (JTA) initiatives revealed that the combination of the products resulting from them could be used to assist in the development of a viable technology transition and insertion plan. Furthermore, a harmonized approach could also possibly be implemented that would address the LPD 17's program concerns related to ensuring interoperability.

A case study was initiated, using the DoD TRM as the key instrument, particularly the TRM 3 Step Methodology to investigate the feasibility of developing a risk mitigation and interoperability assurance methodology for the LPD 17 program. Any architectural aids that could be brought to bear at this point in time in the LPD 17 program without disrupting the program activities would be of significant value to the C4ISR Program Manager and the LPD Program Manager.

Given the scale and complexity of the C4ISR Integration Plan, a typical command-control component of the LPD 17 was selected for analysis to establish viability of the TRM approach taken.. Using the DoD TRM, a mapping methodology was developed that will enable comparisons of initially planned, actual and delivered equipment and systems, and emerging technology and products to be captured and compared, as well as related standards. Though not the intent, a by-product that was also obtained was insight into JTA compliance (matching US Navy standards from the ITSG to the most recent JTA version). The mapping technique subsequently focused on domain controller components of the C2 system. For the case study, one example, IT Security Services, was chosen for the mapping methodology. Phase II will use all TRM Services to populate the database at a much larger scale.

LPD 17 Background

The LPD 17 program is a planned *12-ship procurement* that will integrate easily with the existing amphibious ship force structure. Ultimately, the LPD 17 Class program will be the *functional replacement for four Classes of amphibious ships* that have or will have reached the end of their service life.

LPD 17 primary mission: Amphibious Warfare

An amphibious operation is an attack launched from the sea by naval and landing forces embarked in helicopters, landing craft, and amphibious vehicles on a hostile shore. LPD 17 must be able to embark, transport, and land elements of a landing force in an assault by helicopters, landing craft, amphibious vehicles, and by a combination of these methods ([Figure E-38](#)). The combat power of this ship is its embarked Marines and their equipment.

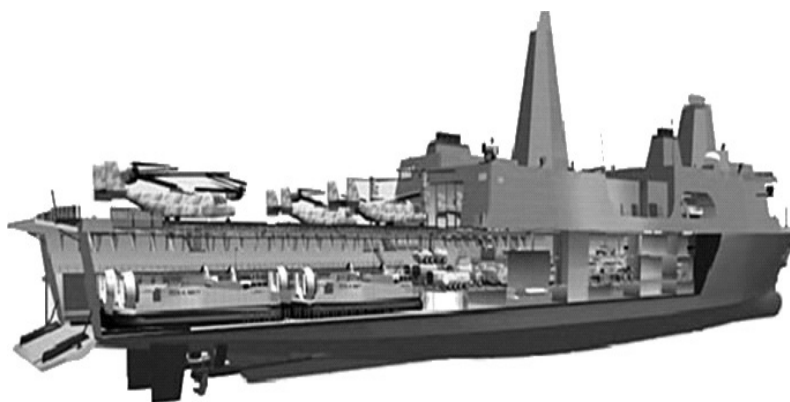


Figure E-38.

LPD 17 MV-22/AAV/UAV Compatible

This is the first ship designed for compatibility with the MV-22 Osprey and the Advanced Amphibious Assault Vehicle. The V-22 is a joint service, multi-mission aircraft with vertical take-off and landing (VTOL) capability. It will also support Unmanned Aerial Vehicles.

The Advanced Amphibious Assault Vehicle (AAAV) is the primary combat vehicle for transporting troops on land and from ship to shore. LPD 17 can carry at least 15 AAAVs along with air cushion landing craft inside its well deck and vehicle stowage areas.

LPD 17 C4ISR Systems

The LPD 17 has a robust Command and Control system that includes: Global Command Control System—Maritime (GCCS-M), Ship Self-defense System MK 2, AN/USG-2 (V) Cooperative Engagement Capability AN/KSQ-1 Amphibious Assault Direction System, AN/UPX 29 IFF System Navigation Sensor System Integration, (NAVSSI) Naval Tactical Command Support System (NTCSS), and the AN/SPQ-14 (V) Radar Display Distribution System.

The LPD 17 Communications system includes: HFRG, UHF, VHF, EHF, SHF, DWTS, and Challenge Athena, Link 11, Link 16, JTIDS, SI Comms, ADNS, TADIX A and an Integrated Voice System.

LPD 17 C4ISR Interoperability

LPD 17 has several areas in which interoperability concerns must be addressed:

- Establishing and maintaining communications between joint services, multi-national forces, and synergism between the Navy-Marine team.
- Amphibious operations are the most complex of naval operations due to timing of events, multiplicity of players, and communications between the personnel and various equipment. For example: ship to shore movement will include communications between many platforms such as AAAVs, V-22s, Armored vehicles, AAAVs, Naval Fire Support (NFS) ships, etc.
- Turnkey integration methodology could impact on migration and technology insertion.
- Migration/technology insertion will impact on standards and standards' compliance (e.g., IT-21, ITSG, JTA) over time.

LPD 17 and Technical Reference Model (TRM)

The TRM therefore is being applied to LPD 17 for the following objectives:

- Assess the utility of TRM to LPD 17 interoperability management,
- Assist in future technical architecture development,
- Identification and validation of standards to use and their required application: IT-21; ITSG; and JTA, and
- Identification of technology insertion and transition aids.

TRM: Approach

Due to the multitude of systems, subsystems, and components in the LPD 17 C4ISR architecture, the TRM has been applied to a single subsystem and one component of such for further decomposition. The subsystem selected was a C2 component, chosen particularly for its Global tie and network-centricity. The C2 component was a processing domain element. A typical diagram is presented characteristic of such controllers. (Figure E-39) .

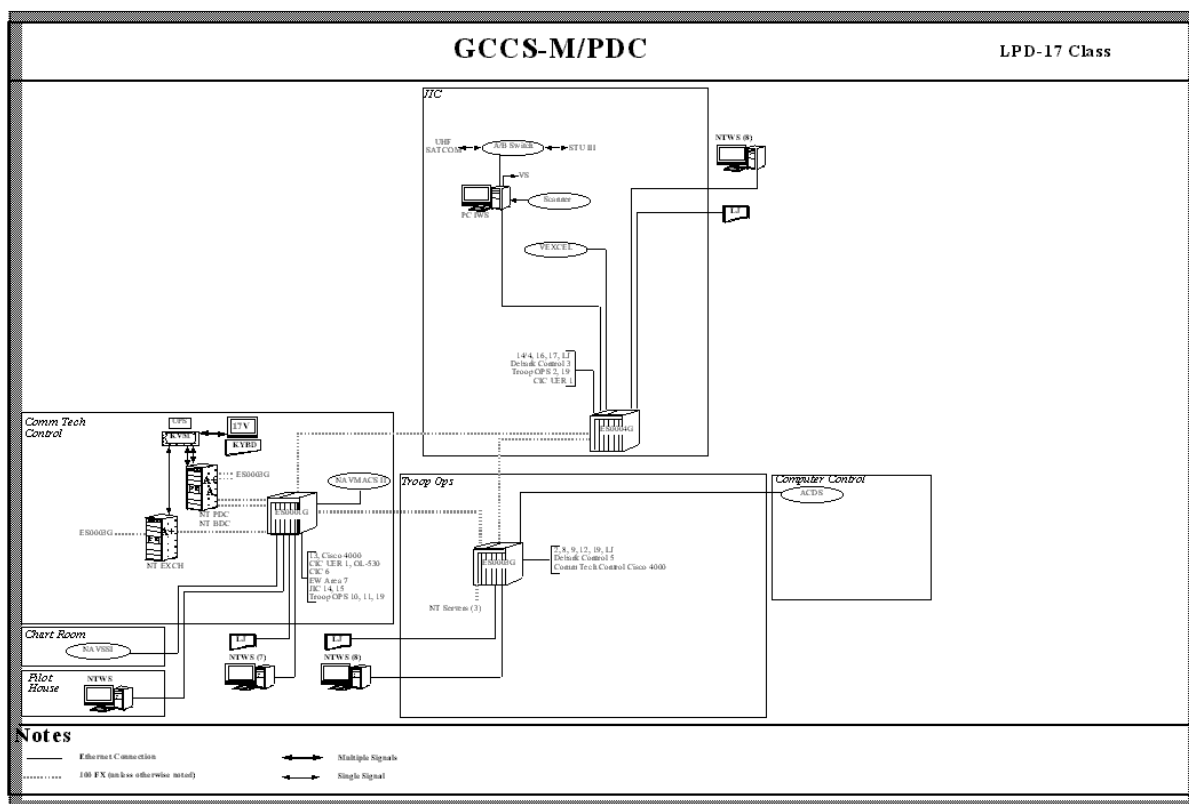


Figure E-39. GCCS-M/PDC Interface Diagram

LPD 17 C4ISR Generic Controller

The C2 system is located throughout the ship in places such as RCS, CIC, Communications Tech Control, JIC, SSES, Troop Ops, CO cabin, etc. The Radio Communication Suite (RCS) consists of different terminals, switching capabilities, and RF sections available to process data off the ship. The C2 system is located in the terminals' section. By itself, the C2 processes data within itself, but it requires input from an antenna or a path to an antenna to emanate or receive the processed data. For connectivity, digital data is developed at a terminal (C2) and then flows through a digital patch panel to either HF, UHF/VHF Line of Site (LOS), UHF SATCOM, EHF SATCOM, SHF SATCOM exciters. The exciters convert the digital data to audio and then to IF (Intermediate Frequency) data and then to RF (radiated frequency) data emitted by an antenna.

The Primary Domain Controller (PDC) component of this subsystem was selected primarily for its role in joint services operations, network centric warfare, as well as for its importance in acting as a gateway to other systems onboard the LPD 17. The PDC's functions are gateway for communications and performing administration. [Figure E-40](#) (PDC: “as-is”) and [Figure E-41](#) (PDC: “to-be”—LPD 17) illustrate technology transition and the impact on functional flow.

Functional Flow Block Diagram : As Is

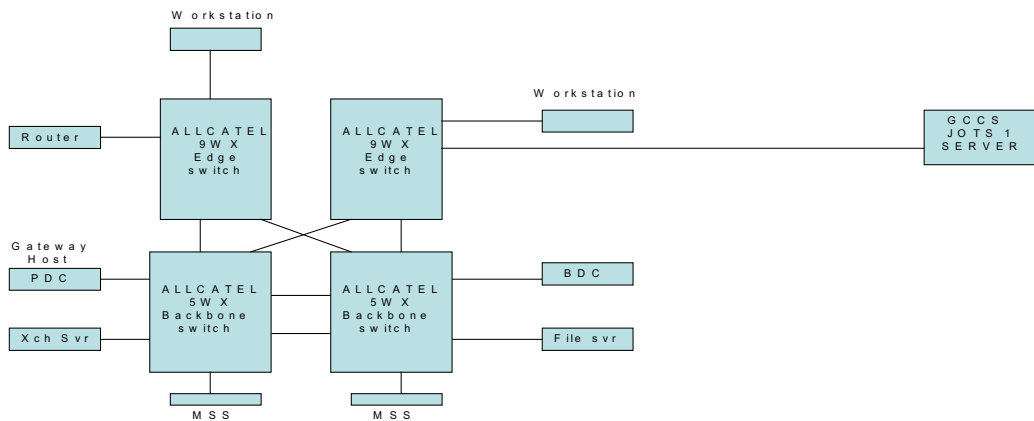


Figure E-40. Current PDC Diagram

Functional Flow Block Diagram - TO BE

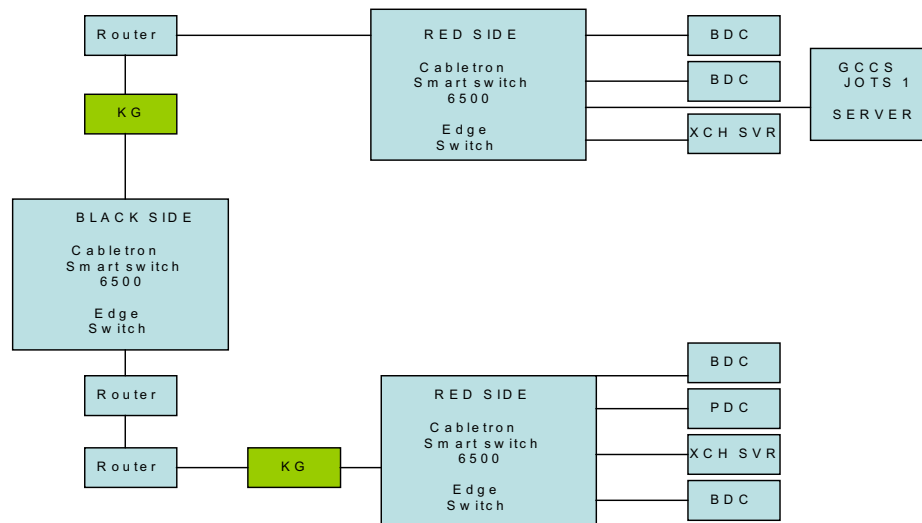


Figure E-41. Future PDC Diagram

TRM Case Study: Process

The specific process which derived from the study follows.

- A. Identify LPD 17 services and interfaces, and map validated services to TRM services.
- B. Map IT21 components (products & standards) to TRM services.
- C. Map ITSG and JTA standards to TRM services.
- D. Analyze and assess common interoperability points for preliminary findings.
- E. Provide recommendations for interoperability solutions.

The results of this process and concluding study (Phase I and Phase II) will provide insight to the Program Manager/Contractor on technology insertion and transition times, as well as facilitate adherence to standards for products installed and future products yet to be installed.

TRM Mapping

The selected methodology allows for a one-to-one mapping of capabilities, services, components and standards. [Figure E-42](#), the mapping foundation below establishes intersecting points for further analysis.

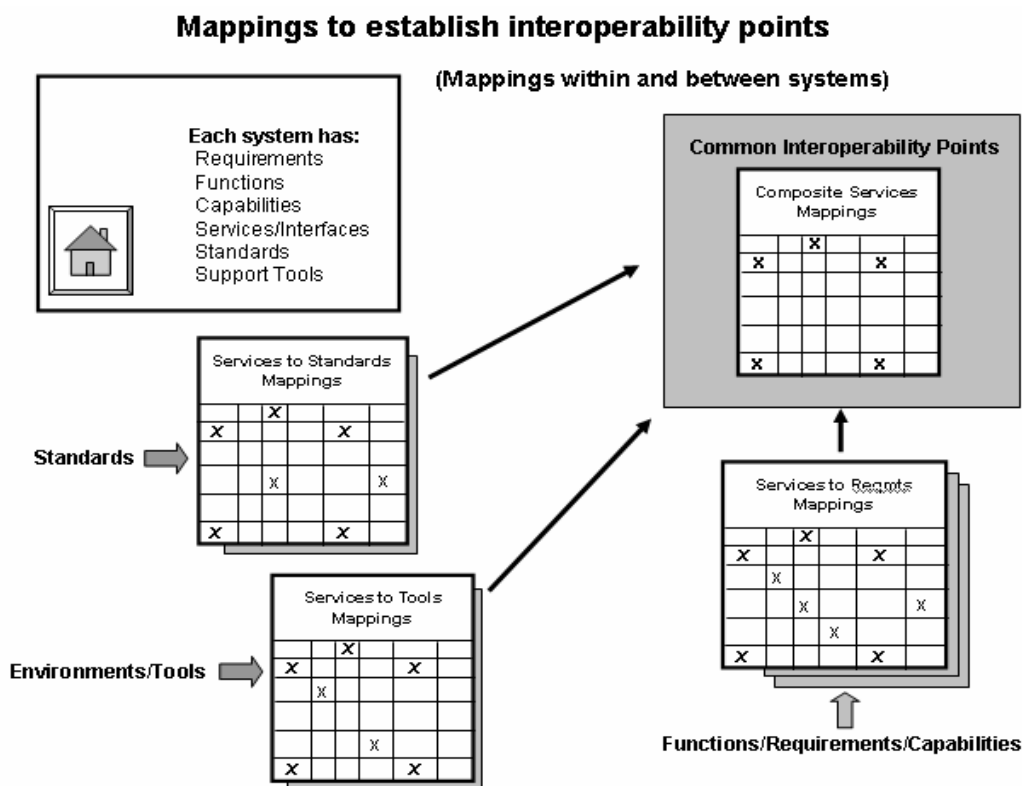


Figure E-42. Mapping Foundation

LPD 17 Mapping

The mapping foundation was used as a baseline for the development of the LPD 17 mapping process in four levels, see [Figure E-43](#). In this case, the possible interoperability points were identified as: TRM services, LPD 17 services, IT21 components, ITSG and JTA standards. Once these categories had been identified, one-to-one mapping then can take place. The methodology establishes an initial deriving of TRM services that are applicable to LPD 17, the matching of those services to existing standards, the viewing of “as-is” and “to-be” configurations, and the merging of the previous mappings in which interoperability areas can be highlighted. Further analysis then can be made for technology insertion, standards compliance, capabilities meeting requirements, etc.

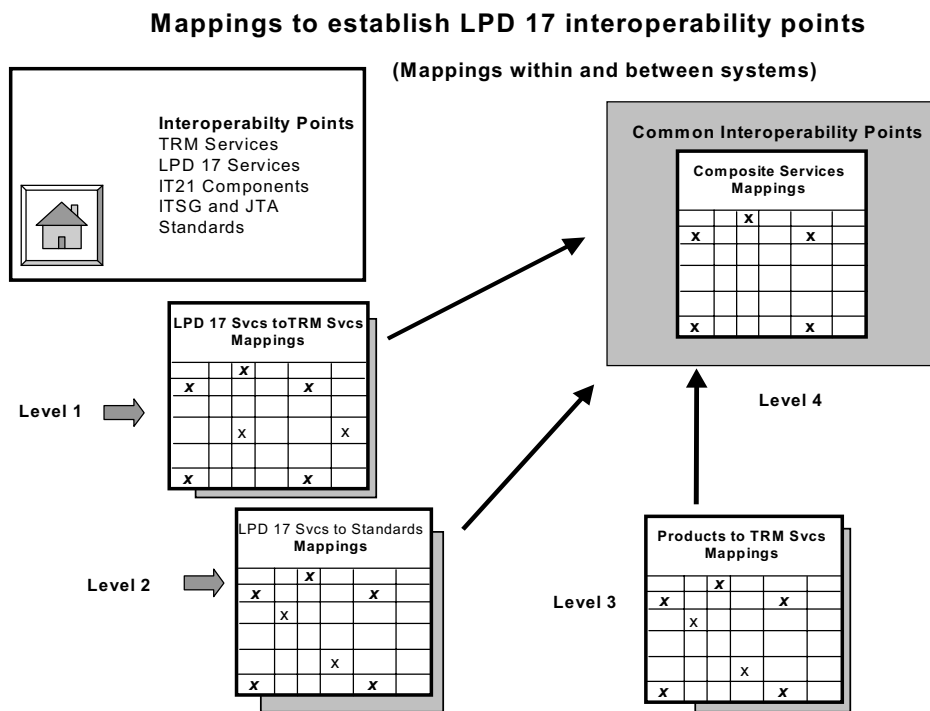


Figure E-43. LPD 17 Interoperability Map

The following section is the specific TRM case study process as applied to the LPD 17 mapping. A matrix format was used in a Microsoft Excel database format.

In the initial steps, a list of services in [Figure E-44](#), was generated from the DoD TRM version 2.0. The services were located in Appendix D of the TRM. Listed were the:

- Entities (note 4.4.1.2 on the slide close up),
- Services (note 4.4.1.2.1 on the slide close up), and
- Sub services (note bulleted items seen on close up of the slide)

Functionality TO Services Matrix
Case Study: LPD-17
System: PDC
Doc Verison: 1.0
Date: 8 July 2002

DOD Version 2.0 TRM SERVICES	Support Application Services (4.4.1.2)
	Multimedia (4.4.1.2.1):
	• Audio Processing
	• Document Processing
	• Electronic Publishing
	• Image Processing
	• Map Graphics
	Communications Applications (4.4.1.2.2):
	• Broadcast
	• Communications Conferencing
	• Enhanced Telephony
	• Organizational Messaging
	• Personal Messaging
	• Shared-Screen Teleconferencing
	• Video Teleconferencing
	Business Processing (4.4.1.2.3):
	• Calendar
	• Calculation
	• Project Management
	• Spreadsheet
	Environment Management (4.4.1.2.4):
	• Batch Processing

Figure E-44. TRM Services

In [Figure E-45](#), SSCC then derived the TRM services utilized by the LPD 17 for the C2 and domain controller system. In doing so, a baseline for systems and their capabilities can then be matched to standards under the ITSG and JTA. This allows for the tracking of technology insertion which can maintain, delete, or add services or capabilities and thus provides for a methodology to assess if compliance is being made to existing standards.

Functionality TO Services Matrix
Case Study: LPD-17
System: PDC
Doc Verison: 1.0
Date: 8 July 2002

DOD Version 2.0 TRM SERVICES	IT 21 DERIVED SERVICES USING DOD TRM 2.0
Support Application Services (4.4.1.2)	
Multimedia (4.4.1.2.1):	
• Audio Processing	
• Document Processing	
• Electronic Publishing	
• Image Processing	
• Map Graphics	
• Multimedia Processing	
• Video Processing	
• Text Processing	
Communications Applications (4.4.1.2.2):	
• Broadcast	
• Communications Conferencing	
• Enhanced Telephony	
• Organizational Messaging	
• Personal Messaging	
• Shared-Screen Teleconferencing	
• Video Teleconferencing	
Business Processing (4.4.1.2.3):	
• Calendar	
• Calculation	
• Project Management	
• Spreadsheet	
Environment Management (4.4.1.2.4):	
• Batch Processing	

Figure E-45. Derived LPD 17 Services

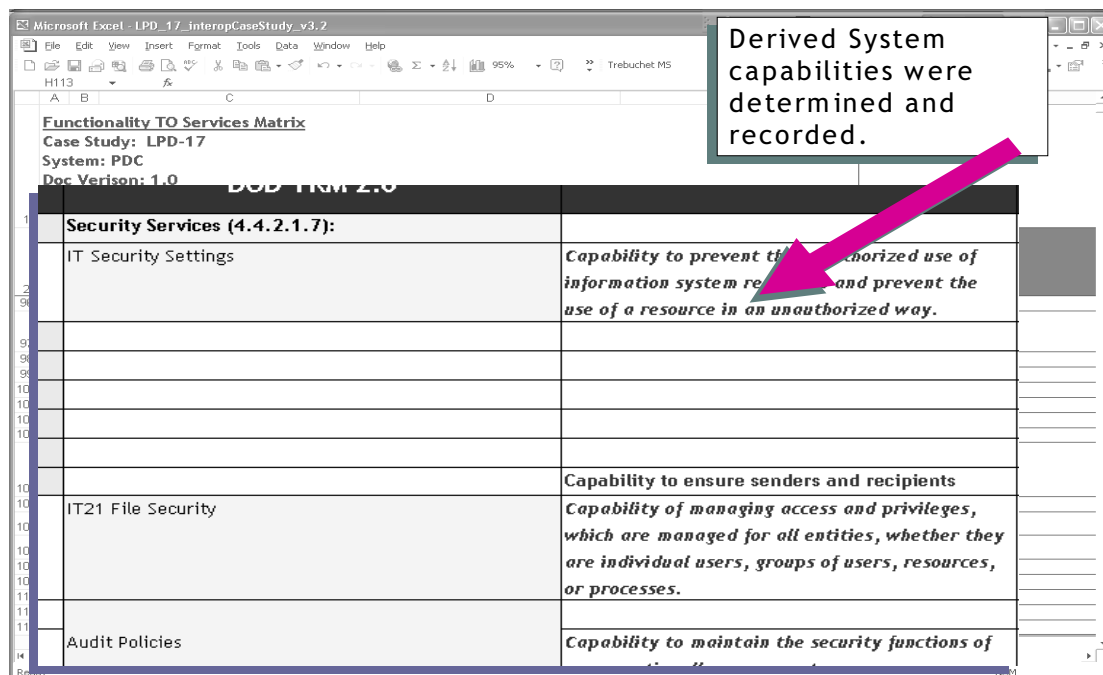
Once the TRM services were identified, these items were separated into a unique column in [Figure E-46](#), to record the services without extraneous information. This second column also assisted in identifying differences in nomenclature and terminology from system to system, and contractor to contractor. The elements are identified in yellow here in the close up.

Derived TRM Services were identified and placed into separate column.

System	Functionality	Security Services	Derived TRM Services
1	Access Control	IT Security Settings	
2	Architectures and Applications		
96	Authentication		
97	Availability		
98	Confidentiality		
99	Integrity		
100	Non-Repudiation		
101	Security Labeling	IT21 File Security	
102			
103			
104			
105			
106	System Management	Aug Policies	
107			
108			
109			
110	Fault Monitoring		
111	Information System Security Management		
112	Other Management		

Figure E-46. TRM Services Identified

Once the derived services were recorded, capabilities required by the LPD 17's command and control system were listed in [Figure E-47](#). To exploit the hypotheses that the TRM may assist in discovery and early notification of interoperability issues, these capabilities were drawn from the services definitions in the TRM. (v. 2.0). A later on verification will need to be performed on the analysis of this information to ensure accuracy of capability to service mapping.



Microsoft Excel - LPD_17_interopCaseStudy_v3.2

File Edit View Insert Format Tools Data Window Help

H113 95% Trebuchet MS

Functionality TO Services Matrix

Case Study: LPD-17

System: PDC

Doc Version: 1.0

Derived System capabilities were determined and recorded.

DOD TRM 2.0	
Security Services (4.4.2.1.7):	
IT Security Settings	Capability to prevent the unauthorized use of information system resources and prevent the use of a resource in an unauthorized way.
IT21 File Security	Capability to ensure senders and recipients Capability of managing access and privileges, which are managed for all entities, whether they are individual users, groups of users, resources, or processes.
Audit Policies	Capability to maintain the security functions of

Figure E-47. LPD 17 Capabilities

Figure E-48. TRM Services Remapped

Microsoft Excel - LPD_17_interopCaseStudy_v3.2

File Edit View Insert Format Tools Data Window Help

A4 Support Application Services (4.4.1.2)

Functionality TO Services Matrix
Case Study: LPD-17
System: PDC
Doc Version: 1.0
Date: 8 July 2002

IT 21 DERIVED SERVICES USING DOD TRM 2.0

REQUIRED SYSTEM CAPABILITIES

Security Services (4.4.2.1.F)

IT Security Settings

Capability to prevent the unauthorized use of information system resources and prevent the use of a resource in an

X

IT21File Security

Capability to ensure senders and

Capability of managing access and privileges, which are managed for all entities, whether they are individual users, groups of users, resources, or

Audit Policies

Capability to maintain the security functions of an operationally secure system.

Audit Directories, files

System Management Services (4.4.2.1.B)

RPC Config

Object Oriented DBMS

Capability to provide facilities and

Suggested_Template_Format / GOT's Delta Install_PDC / Mapping_Template_Format /

Matrix mapping begins with the derived services and capabilities mapped to TRM 2.0 services.

Figure E-49. Level 1 Mapping

LEVEL 2 of the matrix is shown in [Figure E-50](#): FIRST the services and capabilities of the LPD 17 are matched to applicable JTA STANDARDS and added in their own column to the VERTICAL axis of the matrix, THEN the applicable JTA STANDARDS (on the VERTICAL axis) are mapped to the TRM Services on the Horizontal axis.

Microsoft Excel - LPD_17_interopCaseStudy_v3.2

File Edit View Insert Format Tools Data Window Help

80% 100% 120%

Reply with Changes... Ept Review... Trebuchet MS 10

Type a question for help

Functionality TO Services Matrix
Case Study: LPD-17
System: PDC
Doc Version: 1.0
Date: 8 July 2002

IT 21 DERIVED SERVICES USING DOD TRM 2.0	REQUIRED SYSTEM CAPABILITIES	JTA 4.0 STANDARDS	TRM 2.0 SERVICES
Security Services (4.4.2.1.7)	Capability to prevent the unauthorized use of information system resources and prevent the use of a resource in an unauthorized way.	ISO/IEC 15408 Title: Evaluation Criteria for IT Security (parts 1 through 3), 1 December 1999, also documented with the same technical content in Common Criteria (parts 1 through 3), Version 2.1...	
IT21 File Security	Capability to ensure senders and recipients, which are managed for all entities, whether they are individual users, groups of users, resources, or		
Audit Policies	Capability to maintain the security functions of an operationally secure system.		
System Management Services (4.4.2.1.8)			
RPC Config			

Map Standards that support the derived services and capabilities to TRM 2.0 services.

Figure E-50. Level 2 Mapping

In [Figure E-51](#), LEVEL 2 mapping continues with the insertion of standards which were present at contract awarding. This indicates the evolution of standards during the course of the project AND may assist in identifying conflicting or changing standards.

Microsoft Excel - LPD_17_interopCaseStudy_v3.2

File Edit View Insert Format Tools Data Window Help

CM115 80% 100% 120%

Reply with Changes... Ept Review... Trebuchet MS 10

Type a question for help

Functionality TO Services Matrix
Case Study: LPD-17
System: PDC
Doc Version: 1.0
Date: 8 July 2002

IT 21 DERIVED SERVICES USING DOD TRM 2.0	REQUIRED SYSTEM CAPABILITIES	PRECEDING JTA STANDARDS	JTA 4.0 STANDARDS	TRM 2.0 SERVICES
Security Services (4.4.2.1.7)	Capability to prevent the unauthorized use of information system resources and prevent the use of a resource in an unauthorized way.	DoD 5200.28-STD TITLE: DoD Trusted Computer System Evaluation Criteria, December 1985	ISO/IEC 15408 Title: Evaluation Criteria for IT Security (parts 1 through 3), 1 December 1999, also documented with the same technical content in Common Criteria (parts 1 through 3), Version 2.1...	
IT21 File Security	Capability to ensure senders and recipients, which are managed for all entities, whether they are individual users, groups of users, resources, or			
Audit Policies	Capability to maintain the security functions of an operationally secure system.			
System Management Services (4.4.2.1.8)				
RPC Config				

Map standards present at contract commencement.

Identify conflicting or updated standards.

Figure E-51. Level 2 Mapping Continues

LEVEL 2 Mapping as shown in [Figure E-52](#), continues with additional standards mapped to project requirements. Here you will see that the LPD 17 was subject to ITSG standards in the initial requirements specification.

Functionality TO Services Matrix
Case Study: LPD-17
System: PDC
Doc Version: 1.0
Date: 8 July 2002

IT 21 DERIVED SERVICES USING DOD TRM 2.0	REQUIRED SYSTEM CAPABILITIES	PRECEDING JTA STANDARDS	JTA 4.0 STANDARDS	ITSG STANDARDS
IT Security Settings	Capability to prevent the unauthorized use of information system resources and prevent the	DoD 5200.28-STD, TITLE: DoD Trusted Computer System Evaluation Criteria, December 1985	ISO/IEC 15408, Title: Evaluation Criteria for IT Security (parts 1 through 3), 1 December 1999, also documented with the same technical content in Common Criteria (parts 1 through 3), Version 2.1..	DoD 5200.28-STD, TITLE: DoD Trusted Computer System Evaluation Criteria, December 1985
IT21File Security	Capability to ensure senders			
Audit Policies	Capability of managing access and privileges, which are managed in an			
Audit Directories, files	operationally secure system			
System Management Services (4.4.2.1.8)				
RPC Config				

Figure E-52. Level 2 Mapping Continues

In [Figure E-53](#), LEVEL 3 mapping shows documents hardware/software products and configurations of an initial or “as-is” configuration. Products are first aligned on the vertical axis to correspond to services and capabilities. Then (as in the standards mapping) the products are mapped to TRM Services on the horizontal axis.

Functionality TO Services Matrix
Case Study: LPD-17
System: PDC
Doc Version: 1.0
Date: 8 July 2002

IT 21 DERIVED SERVICES USING DOD TRM 2.0	REQUIRED SYSTEM CAPABILITIES	HW/SW PRODUCT(S) SELECTED TO SUPPORT CAPABILITY AS-IS Configuration	TRM SERVICES
IT Security Settings	Capability to prevent the unauthorized use of information system resources and prevent the use of a resource in an unauthorized way.	Windows NT 3.0	
IT21File Security	Capability to ensure senders and		
Audit Policies	Capability of managing access and privileges, which are managed for all entities, whether they are individual users, groups of users, resources, or		
Audit Directories, files	Capability to maintain the security functions of an operationally secure system		
System Management Services (4.4.2.1.8)			
RPC Config			

Figure E-53. Level 3 Mapping

LEVEL 3 continues the mapping in [Figure E-54](#) with the planned/“end state” or “to-be” products. Products are aligned on the vertical axis and mapped to the TRM services on the horizontal. This column is representative of a “Point-in-time” view and additional columns can be added to view transitions over the course of the technology insertion.

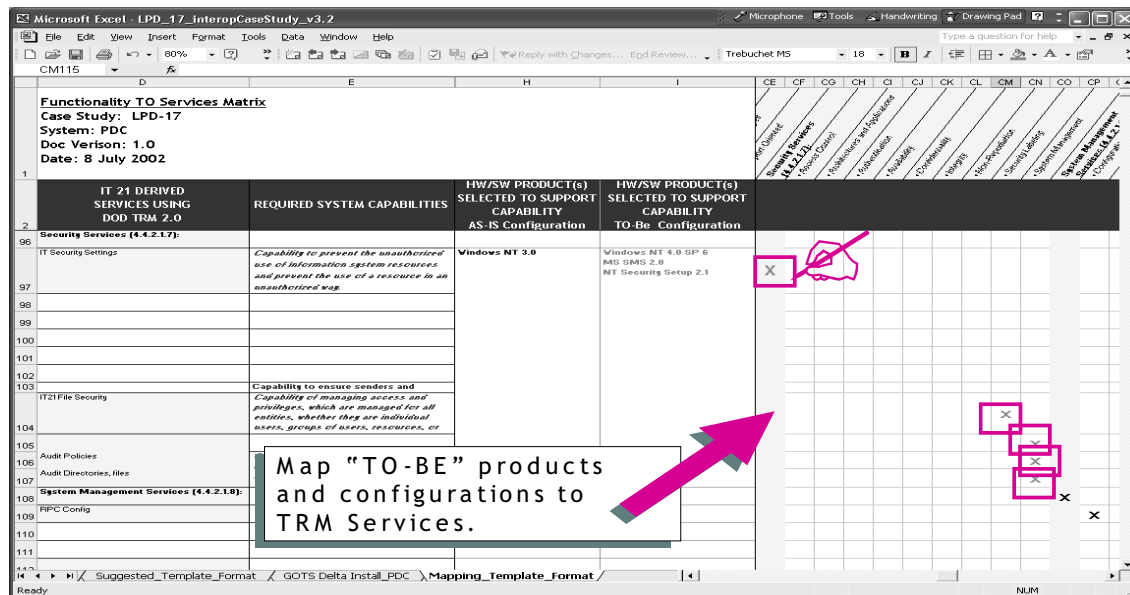


Figure E-54. Level 3 Mapping Continues

LEVEL 4 mapping as shown in [Figure E-55](#), is the alignment that is achieved as a result of merging the previous mappings. Points of interoperability are noted with the following:

- P denotes products issues with interoperability
- S denotes standards issues with interoperability
- U denotes unsupported services
- A check mark indicates all points of interoperability in the system have been examined and no issues exist at present time.

Functionality TO Services Matrix
Case Study: LPD-17
System: PDC
Doc Version: 1.0
Date: 8 July 2002

IT 21 DERIVED SERVICES USING DOD TRM 2.0	REQUIRED SYSTEM CAPABILITIES	CF	CG	CH	CI	CJ	CK	CL	CM	CN	CO	CP	CQ	CR	CS	CT	CU	CV	CW	CX	CY	CZ	DA	DB	DC
Security Services (4.4.2.1.7)																									
IT Security Settings	Capability to prevent the unauthorized use of information system resources and prevent the	✓																							
IT21 File Security	Capability to ensure senders																								
	Capability of managing access and privileges, which are managed for all entities, whether they are individual users, groups of users, resources, or																								
Audit Policies	Capability to maintain the security functions of an operationally secure																								
Audit Directories, files	system																								
System Management Services (4.4.2.1.8)																									
RPC Config																									

Figure E-55. Level 4 Mapping

Preliminary Findings

- ITSG to JTA standards mapping identified both differences in existing standards and evolution of standards over time (JTA 1.0 –JTA 4.0).
- Future projects may result in utilization of interoperability management matrix to monitor systems development through project lifecycle.
- Existing documentation uses incongruent nomenclature to indicate the identical or equivalent functions, services, and standards. (e.g., lack of a common technical reference model)
- Different hardware and software can support the same services, however the implementation can differ: services view (i.e., Operating System Services) vs. implementation is different (i.e., NT vs. UNIX).
- Track and capture “point-in-time” views of the system can be made for comparison.

Summary

- LPD 17 C4ISR interoperability is ongoing.
- The potential for using the TRM to address LPD 17 C4ISR interoperability is promising.
- “As-Is” state during the project lifecycle can be compared to “To-Be” or “end state.”
- The TRM offers a good baseline process for accurate mapping of requirements, services, interfaces, and standards.
- TRM methodology provides insight for point-in-time technology insertion and transition
- TRM may facilitate compliance with a technical architecture (e.g., DoD JTA) between government, contractors, and developers.

Page intentionally left blank.